

TEKNILLINEN KORKEAKOULU

Sähkö- ja tietoliikennetekniikan osasto

Jaakko Salminen

Ohjelmistoprojektin sitoumuksiin vaikuttavat tekijät

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi
diplomi-insinöörin tutkintoa varten Espoossa 24.05.2002

Työn valvoja

Professori Reijo Sulonen

Työn ohjaaja

TkT Jyrki Kontio

Tekijä:	Jaakko Salminen	
Työn nimi:	Ohjelmistoprojektin sitoumuksiin vaikuttavat tekijät	
Päivämäärä:	24.5.2002	
Osasto:	Sähkö- ja Tietoliikennetekniikan osasto	
Professuuri:	Tik-76	Sivumäärä: 66
Työn valvoja:	Professori Reijo Sulonen	
Työn ohjaaja:	TkT Jyrki Kontio	
<p>Ohjelmistoprojektien osapuolet tekevät toisilleen projektista sovittaessa ja sen aikana monenlaisia ja -tasoisia sitoumuksia. Näitä sitoumuksia ei tavallisesti hallita omana alueenaan, vaan ne käsitellään implisiittisesti, osana muuta projektin hallintaa. Tämä tapa hallita sitoumukset vaatii kuitenkin projektia hallitsevilta henkilöiltä varsin pitkää kokemusta. Kun projektista ovat sopimassa ja sitä hallitsevat muut kuin kaikkein kokeneimmat henkilöt, projektiin osallistuvat tarvitsevat sitoumusten hallintaan ohjeistuksen.</p> <p>Tämän diplomityön tarkoitus on selvittää, millaiset tekijät vaikuttavat erilaisissa ohjelmistoprojekteissa tehtäviin sitoumuksiin. Tätä varten työssä luodaan projekteille luokittelu, jonka perusteella projektin henkilöiden on helpompi suuntautua tärkeimmiksi havaittujen sitoumusten hallintaan. Tämän luokittelun pohjalta voidaan myös luoda ohjeisto, joka sisältää heuristiikkaa sitoumusten hallintaan eri tyyppisissä projekteissa.</p> <p>Sitoumuksiin vaikuttavat tekijät tutkittiin tässä työssä toisaalta kirjallisuuskatsauksen, toisaalta empiirisen tutkimuksen avulla. Empiirisessä tutkimuksessa selvitettiin kuudessa eri tyyppisessä projektissa tehdyt sitoumukset peilaamalla niiden sopimuksia ja projektipalaverien materiaalia kirjallisuuskatsauksen perusteella tehtyihin suosituksiin. Tehtyjä havaintoja verrattiin subjektiivisiin kommentteihin projektin hyvistä ja huonoista puolista, ja vertailun perusteella pääteltiin sitoumusten hallinnan vaikutukset projektiin.</p>		
Avainsanat:	Ohjelmistoprojekti, projektisopimus, sitoumukset, sitoumusten hallinta	

Author:	Jaakko Salminen	
Name of thesis:	Commitment factors in software projects	
Date:	24.5.2002	
Department:	Department of Electrical and Communications Engineering	
Professorship:	Tik-76	Pages: 66
Supervisor:	Professor Reijo Sulonen	
Instructor:	Dr Jyrki Kontio	
<p>Parties of a software project make various commitments to each other when negotiating and running a project. These commitments are usually not managed as a separate interest area, but implicitly as part of general project management. This way of managing commitments places big demands on the experience of the persons managing the project. When the most experienced resources are not available, the participants will need guidance in commitment management.</p> <p>The purpose of this thesis is to investigate which factors affect commitment management the most in various types of projects. For this purpose, a classification of projects is created to aid project participants in directing their attention to defining and managing the most important commitments. Also, based of this classification a set of guidelines can be created, which contains heuristics for managing commitments in various types of projects.</p> <p>The factors affecting commitments were researched both by a literature survey and by empirical research. In the empirical research the project agreements and meeting minutes were projected against a set of recommendations based on the literature survey. These observations were compared with a set subjective comments on the good and bad of the projects, and the effect of commitment management was deducted from this.</p>		
Keywords:	Software project, project contract, commitments, commitment management	

Alkulause

Ohjaaja TkT Jyrki Kontio on antanut arvokasta apua ja ohjausta tässä diplomityössä varsinkin sen rakenteeseen ja aiheen käsittelytapaan. Hän myös suuntasi tekijän oikeille raiteille diplomityön teon alussa, mistä hänelle erityinen kiitos.

Valvoja Professori Reijo Sulonen on diplomityön valvonnan ja oikeaan suuntaan ohjaamisen lisäksi antanut tärkeää inspiraatiota ja rohkaisua, mistä hänelle monet kiitokset.

Erityisesti haluan antaa lämpimät kiitokset vaimolleni Leena Mörttiselle, jonka vankkumaton usko ja tuki piti projektini hengissä monista vastoinkäymisistä huolimatta.



Jaakko Salminen

Sisällysluettelo

1	Johdanto	1
1.1	Motivaatio.....	1
1.2	Diplomityön tausta.....	3
1.3	Tavoitteet.....	4
1.4	Tutkimusalue.....	4
1.5	Diplomityön rakenne.....	5
2	Tutkimusmenetelmät	6
2.1	Yleistä.....	6
2.2	Kirjallisuus.....	6
2.3	Tapaustutkimukset.....	6
3	Sitoumukset ohjelmistoprojekteissa: tutkimus ja kirjallisuus	7
3.1	Sitoumusten hallintamalli.....	8
3.2	Ohjelmistokehityksen prosessimallit.....	11
3.3	Projektin neuvottelut.....	16
3.4	Ohjelmistosopimukset.....	21
3.4.1	Ohjelmistosopimuksen vakiopohjat.....	26
3.1	Ohjelmistoprojektin hallinta.....	28
4	Empiirinen tutkimus	31
4.1	Empiirisen tutkimuksen tavoitteet.....	31
4.2	Tutkittavien projektien valinta.....	31
4.3	Käytetyt tutkimusmenetelmät.....	31
4.4	Empiirisen tutkimuksen ongelmat ja luotettavuus.....	32
4.5	Projektien luokittelu	32
5	Tutkimuksen analyysi	34
5.1	Tekijöiden vaikutukset sitoumusten hallintamallin mukaisesti.....	34
5.1.1	Projektin koko.....	34
5.1.2	Osapuolten lukumäärä ja organisoituminen.....	35
5.1.3	Ulkopuolinen vai sisäinen toimittaja?.....	37
5.1.4	Projektin aikataulu.....	38
5.1.5	Liiketoimintamalli.....	39
5.1.6	Valmiskomponenttien käyttö.....	42
5.1.7	Uudelleenkäyttö.....	44
5.1.8	Teknologioiden tuntemus.....	45
5.2	Tekijöiden vaikutus eri tyyppisissä projekteissa.....	47
6	Tutkittavien projektien analyysi	51
6.1	Projekti A	51
6.2	Projekti B	53
6.3	Projekti C	56
6.4	Projekti D	57
6.5	Projekti E	59
6.6	Projekti F	61
6.7	Yhteenveto	63
7	Johtopäätökset ja suositukset	64
7.1	Sitoumukset osana myyntiprosessia.....	64
7.2	Sitoumukset osana projektinhallintaa.....	64
7.3	Jatkotutkimuksen aiheita.....	65
8	Yhteenveto	66

Käytetyt symbolit, lyhenteet ja termit

ATK	Automaattinen tietojenkäsittely
IT	Informaatioteknologia
SEI	Software Engineering Institute
ISO	International Standards Organisation
CMM	Common Maturity Model
4GL	4 th Generation Language
UP, RUP	Unified Process, Rational Unified Process
UML	Universal Modelling Language
LCO	Life Cycle Objective
IPR	Intellectual Property Rights
COTS	Commercial Off The Shelf - valmisohjelmisto
MOTS	Modifiable Off The Shelf – muokattava valmisohjelmisto
IEEE	Institute of Electric and Electronic Engineers
ACM	Association for Computing Machinery
PRINCE2	Projects IN Controlled Environment, version 2

<i>Commitment</i>	Katso Sitoumus
<i>Commitment Specification</i>	Katso Sitoumusten määrittelyt
<i>Sitoumus</i>	Osapuolen projektista sovittaessa ja sen aikana tekemä lupaus, tavoite yms.
<i>Sitoumusten määrittely</i>	Sitoumusten kokoaminen ja kirjaaminen eri osa puolten kesken
<i>Sitoumusten hallinta</i>	Sitoumusten kokoaminen, päivittäminen, katselmointi ja muu niiden ylläpidon toimenpide

1 Johdanto

1.1 Motivaatio

Vielä 10-15 vuotta sitten uusien yritysjärjestelmien tekeminen oli yksinkertaista. Joltain laitetoimittajalta hankittiin riittävän kapasiteetin omaava laitteisto, toiselta toimittajalta siihen varusohjelmat. Tämän jälkeen käynnistettiin yrityksen ATK-osaston kehitystiimissä projekti uuden ohjelmiston tuottamiseksi. Projektin tuloksena saatiin 1-3 vuoden päästä uusi tilausten käsittelyn, kirjanpidon tai asiakashallinnan järjestelmä. ATK-järjestelmät olivat pääasiassa tarkoitettu olemassaolevien prosessien automatisointiin yrityksen sisällä, eikä ulkoisista käyttäjistä tai yhteyksistä tarvinnut välittää. Tietojenkäsittelyn järjestelmillä ei haettu suoraa kilpailuetua, vaan oman toiminnan tehostamista.

Nykyinen tapa käyttää IT-järjestelmiä on perustavalla tavalla erilainen kuin aikaisemmin. Järjestelmillä ei ainoastaan tehosteta omaan toimintaa, vaan usein ennen kaikkea haetaan selkeää kilpailuetua kilpailijoihin nähden tuomalla sekä uusia palveluja asiakkaitten käyttöön että mahdollistamalla uusien liiketoiminnan muotojen käyttöönottoa mm. uudenlaisten laskutustapojen, asiakaspalvelun tai kumppanuksien muodossa. IT-järjestelmä on muuttunut sisäisestä työkalusta liiketoiminnan keskeiseksi katalysaattoriksi. Monet varsinkin ns. uuden talouden yrityksistä jopa tuovat palvelunsa asiakkaalle ainoastaan IT-järjestelmien kautta, ja yrityksen tietojärjestelmien kehityksen positiiviset tai negatiiviset suunnat heijastuvat välittömästi yritysten pörssikursseihin.

Nykyinen kehitys on johtanut siihen, että perinteinen tapa kehittää tietojärjestelmiä ei enää pysty vastaamaan uusiin tarpeisiin. Ohjelmistojen kehityksen on oltava nopeaa, koska uudet järjestelmät on saatava käyttöön ennen kuin liiketoimintamahdollisuuden ikkuna sulkeutuu. Toiminnan tehostamiseen ja oman ydinosaamisensa kehittämiseen keskittyminen on karsinut suuret kehitysosastot ulos yritysten organisaatioista, joten ohjelmistojen kehityshankkeita käynnistetään alihankkijoiden ja kumppaneiden kanssa. Ohjelmistojen kehityksessä pyritään hankkimaan valmisohjelmistoja niin kattavasti kuin se on järkevää, ja teettämään niiden sovitustyöt ja uuden kehittäminen ulkoisilla toimittajilla.

Tarve lyhentää ohjelmistojen kehityksen syklejä on tuonut esille myös jatkuvasti kasvavan tarpeen luoda ohjelmistoja uudelleenkäytettävistä komponenteista. Tässäkin mielessä ohjelmistotyö lähestyy perinteistä teollisuutta tai rakennusalaan, joissa vakiokomponenttien käyttö on jo pitkään ollut tavallinen tapa toimia.

Tämä tavoite edelleen lisää ohjelmistoprojektin sopimusten merkitystä. Seuraava ohjelmistokomponentin käyttäjä saattaa olla kokonaan eri organisaatiosta kuin sen alkuperäinen kehittäjä, ja hänen on luotettava määritelyihin rajapintoihin ja käyttäytymiseen. Määrittelemällä ohjelmistoprojektin sopimus tarkoituksenmukaisesti, ohjelmistojen tilaaja voi paremmin varmistua siitä, että komponentti on hänen organisaatiossaan määritelyjen standardien mukainen.

Samalla kun ohjelmistojen kehitys on useimmilla käyttäjillä lähes kokonaan ulkoistettu, on projekteihin osallistuvien osapuolten lukumäärä kasvanut, ja yrityksistä muodostunut verkostoja. Ohjelmistoprojektit ovat alkaneet

muistuttaa rakennusalan projekteja: Asiakas tilaa työn yhdeltä toimittajalta, joka edelleen käyttää useita alihankkijoita ohjelmiston eri osien tekemiseen. Asiakas tekee sopimuksen integroivan osapuolen kanssa, joka edelleen tekee osaprojekteista sopimukset alihankkijoiden kanssa. Ketju saattaa jatkua jopa pidemmälle, koska alihankkijat saattavat edelleen ostaa tai tilata ohjelmiston osia muilta toimittajilta. Tällaisen verkostoitumisen ansiosta projektisopimuksia tehdään projekteissa monella tasolla, ja varsinkin integroivan toimittajan on syytä panna paljon painoa sopimusten kattavuudelle tarkoituksenmukaisuudelle.

Tällaisessa tilanteessa sekä projektin hallinta että projektisopimusten tekeminen on haasteellista. Jotta projektisopimuksen tekeminen onnistuisi riittävän hyvin, on kattavan sopimuksen tekeminen oltava mahdollisimman suoraviivaista ja helppoa. Monissa yrityksissä onkin tehty paljon työtä toimitusehtojen, vastuukysymysten, tekijänoikeudellisten kysymysten ja muiden projektin suorituksen aikana ja sen jälkeen vastaan tulevien kysymysten hallintaan, ja yleisesti ottaen tämä puoli ohjelmistoprojekteista on hyvin hallinnassa. Samoin itse ohjelmistotyön teknisten tavoitteiden ja sisällön määrittelemine on hyvin kartoitettu alue. Sen sijaan projektin sisältöön vaikuttavien ei-teknisten asioiden määrittelyyn ei ole monessakaan yrityksessä panostettu.

Projektien perinteiset menestystekijät eivät kuitenkaan ole muuttuneet, pikemminkin niiden merkitys on korostunut. Projektin myöhästyminen voi aiheuttaa parhaan tilaisuuden menettämisen ja yli budjetin päässeet kustannukset voivat tehdä projektista taloudellisesti kannattamattoman. Projektin onnistunut hallinta, pätevät työntekijät, avainhenkilöiden saatavuus tarvittaessa ja muut tärkeäksi havaitut tekijät ovat edelleen välttämättömiä edellytyksiä onnistuneelle projektille.

Tärkeimpiin tekijöihin kuuluu myös projektin määrittelemine hyvin ennen kuin varsinainen tekeminen aloitetaan. Ohjelmistoprojektin sopimus onkin vähintään yhtä tärkeä menestystekijä kuin yllä mainitut, mutta koska sen tekemisestä on helpompi tinkiä kuin aikaisemmin mainituista tekijöistä, se on muodostunut monessa projektissa merkittäväksi riskitekijäksi. Jos eri osapuolilla ei ole riittävän tarkkaa ja ennen kaikkea yhtenäistä kuvaa mm. projektin tavoitteista, eri osapuolten tehtävistä, tai projektin liittymisestä muuhun liiketoimintaan, otetaan erittäin suuri riski.

Kattavasti tehdyt projektin taustan määrittelyt ovat tärkeitä projektin kaikille osapuolille. Tilaaja pystyy paremmin varmistamaan projektin tulosten tarkoituksenmukaisuuden ja oikeellisuuden, kun myös toimittajat ovat perillä yrityksen liiketoiminnallisista tavoitteista. Toimittaja pystyy puolestaan paremmin arvioimaan omat mahdollisuutensa projektin toteuttamiseen taloudellisesti järkevällä tavalla. Hän pystyy arvioimaan omien resurssiensa sopivuuden, projektin sopivuuden yleisten linjauksiensa kanssa ja asetettujen aikataulujen ja muiden reunaehtojen vaikutuksen projektin toimitukselle. Hyvin määritellyt sopimuksen sisältö ja ehdot ovat jopa enemmän toimittajan etu kuin tilaajan. Toimittajan tarkka käsitys projektista toimii myös tilaajan eduksi, koska näin hän saa varmemmin sellaisen toteutuksen, josta on hänelle hyötyä.

Miksi ohjelmistoprojektit sitten epäonnistuvat? Zeiger tiivistää tavallisimmiksi syiksi osaamisen yliarvioinnin, sisäiset sodat ja projektin

hallinnan, joka ei sovi liiketoiminnallisiin tavoitteisiin [Man99]. Osaamisen yliarviointi tai -arvostus tapahtuu tilanteessa, jossa yhdellä alueella ansioitunut asiantuntija saa tehtäväkseen hänelle vieraan alueen projektin, eikä osaa arvioida kaikkia tehtävän vaatimuksia ja riskejä oikein. Joissain tapauksissa organisaation osat saattavat kilpailla keskenään, jopa tarkoituksella. Tällaisessa tilanteessa keskinäinen kilpailu vie huomion projektin haasteiden voittamiselta tai pahimmillaan johtaa jopa toisten osapuolten tahalliseen harhauttamiseen. Projektin hallinta on ristiriidassa liiketoiminnallisten tavoitteiden kanssa, kun taloudellisten tavoitteiden ja mittarien merkitystä ei tunnusteta tai niitä kierretään.

Abe, Sakamura ja Aiso [Abe79] ovat puolestaan tutkineet yhdentoista ohjelmistoprojektin epäonnistumiseen vaikuttaneita tekijöitä. Yleisin tekijä tutkimuksen mukaan on ollut projektin jäsenten väärinarvioinnit, jotka useimmiten johtuivat henkilöiden kokemuksen puutteesta.

1.2 Diplomityön tausta

Kirjoittaja toimii asiantuntijapalvelujen johtajana ohjelmistotyötä ja konsultointia toimittavassa yrityksessä, Sun Microsystems Oy:ssä. Yritys suorittaa ohjelmistokonsultointia tavoitteena Java- ohjelmistojen arkkitehtuurien kehittäminen asiakkailleen, ja hankkii ohjelmistotyötä projekteihin edelleen omilta alihankkijoiltaan. Kirjoittaja on myös itse osallistunut useiden projektien määrittelyyn ja myyntiin. Projektit ovat olleet luonteeltaan pääosin kokonaisjärjestelmien toimituksia, joissa ohjelmistoprojekti on ollut osana, mutta monissa projekteissa ohjelmistot ovat olleet pääasiallinen sisältö.

Kuten todennäköisesti useimmissa ohjelmistoprojekteja toimittavissa yrityksissä, Sun Microsystems Oy:ssä on havaittu projektin menestyksen paljolti riippuvan siitä, miten hyvin sitoumukset on sovittu osapuolten kesken. Jos yrityksellä ei ole pitkiä perinteistä ohjelmistoprojektien toimittamisesta, kuten on tilanne Sun Microsystems Oy:ssä, sitoumusten määrittelyssä nojataan käytännössä lähes täysin joidenkin konsulttien kokemukseen ohjelmistoprojektin käynnistämisestä ja ohjaamisesta. Lisäksi yhteiset ohjeet ja pohjat ohjelmistosopimusten tekoon puuttuvat.

Monet näkökohdat projektisopimuksen teosta on itse asiassa jo valmisteltu perusteellisesti. Tietotekniikan liitto (TTL ry) Suomessa ja useat vastaavat organisaatiot kansainvälisesti ovat luoneet hyviä pohjia sopimuksille, ja lisäksi varsinkin kansainvälisillä yrityksillä on omia sopimus pohjiaan ohjelmistoprojekteja varten. Näissä sopimus pohjissa on keskitytty erityisesti määrittelemään asiat, jotka ovat tärkeitä toimituksen aikana ja sen jälkeen. Näitä ovat esimerkiksi toimitusehdot, laskutusehdot, tekijänoikeudelliset asiat, ohjelmistojen lisenssit, tarjottujen palvelujen laajuuden rajaukset maantieteellisesti tai muuten, tietojen luottamuksellisuuden varmistaminen, takuut, vastuun rajoitukset, sovellettavat lait, ja keskeytys ehdot. Näistä asioista sopiminen on tärkeää ennen kaikkea riskien rajaamisen kannalta.

Nämä sopimus pohjat ja ohjeistot kattavat varsin hyvin projektin muodolliseen toimittamiseen ja poikkeustilanteisiin liittyvät seikat, mutta eivät auta paljoakaan projektin varsinaisen sisällön määrittämisessä. Projektin tavoitteet on syytä määritellä yhtä huolella, ei ainoastaan teknisessä mielessä, vaan varsinkin sen tulosten aiottujen vaikutusten osalta.

Jos ohjelmistoprojekti on hyvin määritelty sopimuksessa, sen onnistuminen on todennäköisempää, eikä poikkeustilanteisiin jouduta.

1.3 Tavoitteet

Tämän diplomityön tavoitteena on selvittää, mitkä ovat tärkeimmät ohjelmistoprojektin alussa tehtäviin sitoumuksiin vaikuttavat tekijät, sekä minkälaisia sitoumuksia tehdään missäkin tilanteessa. Tässä työssä käytetään tarkastelun lähtökohtana Kontion kehittämää sitoumusten hallintamallia [Kon98]. Tämän työn yksi tavoite on myös kehittää ko mallia kattamaan tärkeimmät tekijät.

Mallin pohjalta voidaan luoda ohjeisto sekä erilaisten ohjelmistoprojektien sopimusten tekemistä varten että tärkeiden tekijöiden mukaiseen projektin seurantaan. Tätä ohjeistoa varten pyritään luomaan projekteille luokittelu, jonka tarkoituksena on ohjata olennaisimmiksi todettujen sitoumusten hallintaan. Luokittelun pohjana käytetään Sun Microsystems Oy:n asiantuntijapalvelujen tyypillisimpien hankkeiden kuvauksia. Luokittelun ei ole tarkoitus olla tyhjentävä, mutta sen perusteella voidaan kattaa enemmistö nykyisellä strategialla käynnistettävistä hankkeista. Muut hankkeet voidaan tulkita poikkeuksina näistä tyyppitapauksista, ja tulevaisuudessa mahdollisesti tapahtuvien strategiamuutosten myötä voidaan luoda uusia luokkia.

Tulokset voidaan myös esittää matriisina, jonka sarakkeina ovat sitoumusten määrittelyn alueet ja riveinä tutkimuksen tuloksena löytyneet projektin onnistumisen kannalta tärkeät alueet. Kunkin solun sisältö on tutkimuksen tuloksena löytyneiden alueeseen liittyvien kysymysten joukko ja heuristiikkaa kysymysten ratkaisemiseksi.

Koska varsinkin kokeneimmat resurssit ovat useimmissa yrityksissä erittäin kysyttyjä, projektisopimusta ovat usein tekemässä henkilöt, joiden oma tuntuma sopimusten tekemisessä ei ole kovin laaja. Jotta kaikissa tapauksissa saavutettaisiin sopimus, joka on sekä riittävän kattava että tavoitteiltaan järkevä, on aiheellista luoda puitteet, jotka auttavat tuohon tavoitteeseen pääsemisessä. Tällaiset puitteet toimivat myös kokeneempien henkilöiden tukena ja varmistavat, ettei mikään olennainen osa sovittavista asioista jää puuttumaan.

Eräänlaisena sivutuotteena työssä syntyy myös katsaus siitä, mitä erityisalueita on tähän mennessä tutkittu, ja millaisia tuloksia eri lähteissä on eri erityisalueilla. Jos jonkin erityisalueen on empiirisesti todettu vaikuttavan merkittävästi projektin onnistumiseen, mutta siitä ei kirjallisuudessa ole mainintoja, olisi sitä todennäköisesti tutkittava lisää. Samoin voidaan toimia jos jollain alueella kirjallisuudessa ja empiirisissä selvityksissä on ristiriitaa.

1.4 Tutkimusalue

Ohjelmistoprojektien sitoumuksilla tarkoitetaan tässä työssä niitä panostuksia, tavoitteita ja velvoitteita, joista osapuolet sopivat projektin sopimuksessa tai projektin aikana. Sitoumusten avulla projektin osapuolet voivat luottaa muiden osapuolten panostuksesta projektin tavoitteiden saavuttamiseksi, ja suunnitella omaa toimintaansa sekä puolestaan tehdä

omia sitoumuksia. Sitoumuksia ovat esimerkiksi tiettyjen resurssien allokointi projektin käyttöön tarvittavissa määrin, tiettyjen tuotosten toimittaminen tiettyssä aikataulussa, tai osapuolten strateginen linja projektin ja sen myötä syntyneiden yhteistyömuotojen suhteen.

Tässä työssä keskitytään erityisesti tilanteisiin, joissa ohjelmistoprojektin osapuolet edustavat erillisiä organisaatioita, eli ovat tyypillisesti esimerkiksi asiakas ja toimittaja. Tähän rajaukseen päädyttiin toisaalta siksi, että kirjoittaja on töissä projekteja toimittavassa organisaatiossa, joka siksi pystyy hyödyntämään työn tuloksia, toisaalta siksi, että sitoumukset määritellään tavallisesti paremmin sellaisissa projekteissa, joissa liiketaloudelliset kustannukset ja hyödyt ovat selkeämmin nähtävissä.

Tapausta, joissa ohjelmiston toimittaja on osa samaa yritystä tai organisaatiota kuin tilaaja, voidaan ajatella erikoistapauksena asiakas/toimittaja suhteesta. Usein näissäkin projekteissa pyritään organisaatioiden eri osia käsittelemään mahdollisimman pitkälle siten kuin kyseessä olisivat asiakas ja toimittaja, jotta myös organisaation sisällä voidaan tehokkuutta seurata mahdollisimman hyvin.

1.5 Diplomityön rakenne

Luvussa 2 esitellään diplomityössä käytetyt tutkimusmenetelmät, aiheen käsittelytapa, kirjallisuusmateriaalin valinta, ja tutkittavan materiaalin käsittely.

Luvussa 3 käsitellään tutkimus ja kirjallisuus, joissa käsitellään tai sivutaan sitoumuksiin vaikuttavia aihealueita.

Luvussa 4 kuvataan empiirisen tutkimuksen tavoitteet, tutkittavien projektien valinta, empiirisen tutkimuksen mahdolliset ongelmat ja luotettavuus, sekä esitellään projektien luokittelun tausta ja perusteet.

Luvussa 5 kuvataan kirjallisuuskatsauksen perusteella koottujen sitoumuksiin vaikuttavien tekijöiden analyysi tämän työn pohjana käytetyn sitoumusten hallinnan mallin mukaisesti.

Luvussa 6 analysoidaan tutkittavat projektit tyypeittäin luvun 5 analyysin pohjalta luotuun projektityyppien sitoumushallinnan malliin peilaten.

Luvussa 7 käsitellään tiivistelmänä projektityyppien sitoumushallinnan mallin toimivuus ja soveltuvuus kirjoittajan organisaation käyttöön, ja muut johtopäätökset, sekä hahmotellaan työn aikana esille tulleita jatkotutkimuksen aiheita.

Luvussa 8 käsitellään yhteenvetona työn tulokset.

2 Tutkimusmenetelmät

2.1 Yleistä

Tässä diplomityössä kuvattu työ on kvalitatiivinen tutkimus ohjelmistoprojektien sitoumuksiin ja niihin vaikuttaviin tekijöihin. Vaikka kvantitatiivinen lähestymistapa olisi ollut kiinnostava ja houkuttelevakin, sen vaatimaa laajaa tutkimusaineistoa ei ollut mahdollista koota tätä työtä varten. Lisäksi kvalitatiivisella tutkimuksella saavutettiin ne tavoitteet, jotka työlle asetettiin, eli pystyttiin sekä hankkimaan vahvistusta sitoumusten tähänastiselle teorialle, että tunnistamaan säännöllisyyksiä erilaisten sitoumusten hallinnan tarpeellisuudesta eri tyyppisissä projekteissa.

2.2 Kirjallisuus

Tutkimuksen kohteena ollut alue on toisaalta kosketuspinnaltaan huomattavan laaja että erillisenä aiheena vähän käsitelty. Tästä syystä tässä työssä päädyttiin suhteellisen laajan kirjallisuusselvityksen tekemiseen.

Kirjallisuudeksi valittiin lähteitä, joita on sekä käytetty paljon lähteenä myös muissa tätä aluetta sivuavissa tutkimuksissa, että käytetään paljon käytännön projekteja suunniteltaessa ja johdettaessa. Näin päädyttiin valitsemaan varsinkin Barry Boehmin artikkeleita ja kirjoja, sekä ohjelmistoprojekteja ja niiden hallintaa käsitteleviä lähteitä.

Yhtenä lähteenä käytettiin myös ohjelmistoliiketoiminnasta tehtyä tuoretta väitöskirjaa, joka myös käsittelee erilaisten ohjelmistoyritysten suhdetta asiakkaisiinsa ja sen kautta antaa viitteitä yleisimmistä ja tarpeellisimmista sitoumuksista.

2.3 Tapaustutkimukset

Kirjallisuuskatsauksen lisäksi tutkimukseen valittiin 6 ohjelmistoprojektia, joiden sitoumushallinta tutkittiin vertaamalla niissä tehtyjä sitoumusten määrittelyjä kirjallisuuskatsauksen perusteella tehtyyn listaan sitoumuksiin vaikuttavista tekijöistä. Vertailu perustui pääasiassa projektien kirjalliseen materiaaliin, jota olivat mm. projektien sopimukset, työnkuvaukset, ja projektipalaverien pöytäkirjat.

Kustakin projektista koottiin ensimmäisenä lyhyet listat projektissa hyvin ja huonosti menneistä asioista. Sitoumusvertailun avulla pyrittiin löytämään selitys mahdollisimman monelle näistä kohdista.

3 Sitoumukset ohjelmistoprojekteissa: tutkimus ja kirjallisuus

Tässä luvussa käsitellään niitä ohjelmistoprojektien tutkimuksen alueita, joilla on suurin vaikutus sitoumusten määrittelyyn. Koska tässä työssä on tarkoitus käsitellä kaikkia sitoumuksia, joita projektin aikana tehdään, on tarpeen kattaa sekä ohjelmistoprojektin sopimukseen vaikuttavat alueet, kuten sopimusten tutkimus, vakiosopimukset, ja sopimusten neuvottelujen tutkimus, että projektin aikana vaikuttavat alueet, kuten ohjelmiston kehityksen prosessimallit ja ohjelmistoprojektin hallinnan tutkimus.

Erityisesti ohjelmistoprojektien sitoumuksia ovat tutkineet ainoastaan Kontio, Sulonen ja Pitkänen. Tämä työ käyttää heidän rakentamaansa sitoumusten määrittelyn mallia perusmallinaan ja rakentuu sen päälle.

Sitoumuksia ohjelmistoprojekteissa on tähän mennessä tutkittu vain vähän omana alueenaan ohjelmistotuotannon kirjallisuudessa. Kontio, Pitkänen ja Sulonen ovat tutkineet suomalaisia ohjelmistoprojekteja ja rakentaneet sitoumusten hallintamallin [Kon98]. Muuten sitoumuksia on käsitelty lähinnä joko henkilöstöhallinnan osana, eli henkilöstön sitoutumisena organisaatioon tai projektiin, tai johdon sitoutumisena, jonka tunnustetaan laajasti olevan ratkaisevaa projektin menestykselle. [McC98] Sitoumusten muodollinen määrittely ja neuvottelu on sen sijaan harvoin käsitelty alue.

Ohjelmistoprojektien sopimukset ja suunnitelmat ovat tavallisin perusta sitoumusten määrittelylle. Joitakin standardeja ja esimerkkejä sopimuksista ja projektisuunnitelmista on olemassa, mutta ne eivät kata kaikkia sitoumusten määrittelyn alueita. Projektisuunnitelmissa keskitytään pääasiassa aikataulujen ja kustannustavoitteiden määrittämiseen, prosessien määrittämiseen, ja riskien hallinnan järjestämiseen, mutta eivät kata projektin motiivitekijöitä tai ongelmatilanteiden hallintaa. [IEE87] Ennen kaikkea ne eivät käsittele lainkaan sitoumusten määrittelyn prosesseja.

Riskien hallinnan tutkimus käsittelee sitoumuksia epäsuorasti. SEI:n riskien hallinnan menetelmä olettaa että osapuolet saavuttavat yhteisymmärryksen ja kaikkia riskeihin liittyviä asioita käsitellään avoimesti. Tämä lähestymistapa ei kuitenkaan ota huomioon mahdollisia osapuolten eturistiriitoja, piilotavoitteita, erilaisia näkökantoja. SEI:n malli ei myöskään käsittele riskien hallinnan sopimuksellista näkökulmaa tai siihen liittyviä neuvotteluja.

Barry Boehmin WinWin- malli ottaa riskien hallinnassa huomioon eri osapuolten näkökannat ja tarvittavat neuvottelut.[Boe89] Vaikka tämä on tunnetuin malli osapuolten odotusten käsittelyyn, se rajoittuu tavoitteiden määrittämiseen eikä sisällä muodollista menetelmää sitoumusten määrittelylle.

Kontion RiskIt menetelmä ottaa huomioon eri osapuolet ja heidän erilaiset odotuksensa, ja sisältää menetelmän jäljittää riskit osapuoliin ja heidän tavoitteisiinsa. [Kon97] Tämäkään malli ei käsittele sitoumusten määrittelyä ja neuvotteluja yksityiskohtaisesti.

Tässä diplomityössä tarkastellaan ohjelmistoprojektien sitoumuksia muiden ohjelmistotuotannon alueiden, kuten ohjelmistonkehityksen mallien, ohjelmistosopimusten, ja ohjelmistoprojektin hallinnan kautta.

3.1 Sitoumusten hallintamalli

Tämän diplomityön perustana käytetty sitoumusten hallintamalli [Kon98] on tällä hetkellä ainoa malli ohjelmistoprojektin sitoumusten kuvaamiseen. Siinä on pyritty jäsentämään sitoumukset jakamalla ne neljään alueeseen, jotka ovat keskeisimmät projektin menestyksen kannalta: projektin tavoitteet, prosessien määrittelyt, ja riskien ja ongelmatilanteiden hallinta. Seuraavassa esitetään sitoumusten hallintamallin peruspiirteet, siten kuin ne on Kontion et al artikkelissa kuvattu.

Motiivitekijät ovat ne syyt, miksi projekti ylipäättään halutaan toteuttaa. Nämä ovat Kontion mallissa tyypillisesti liiketaloudellisia etuja kuten myynnin tai tehokkuuden lisääminen, laadun parantaminen tai asiakaspalvelun tehostaminen.

Projektin osapuolet tekevät motiivitekijöiden avulla selväksi toisilleen mitä projektilla viime kädessä tavoitellaan. Ne kuvaavat enemmänkin projektin hengen kuin sen kirjaimen, minkä muut alueet puolestaan kuvaavat. Niiden avulla osapuolet osaavat tulkita projektin tavoitteet ja prosessit samalla tavalla. Lisäksi motiivitekijät auttavat mahdollisten ongelmatilanteiden ratkaisussa: ulkopuoliset tahot kuten oikeusistuimien voivat tarvittaessa niiden avulla ratkaisuisaan ottaa huomioon osapuolten motiivit ja edut.

Joissakin tapauksissa kaikkia motiivitekijöitä ei kuitenkaan ole mahdollista dokumentoida esimerkiksi niiden luottamuksellisuuden takia. Lisäksi on varottava, etteivät motiivitekijät muodostu sitovammiksi kuin oli tarkoitettu. On esimerkiksi mahdollista, että ohjelmistoprojektin tilaaja haluaa, että toimittaja osallistuu tilaajan riskien kantamiseen huonosti määriteltyjen motiivitekijöiden takia, mutta ei saa pystyä hinnoittelemaan korvausta tästä vastuunkannosta.

Projektin tavoitteet määrittelevät milloin ja mitä projektin on tarkoitus konkreettisesti tuottaa, ja millaisilla rajoitteilla, kuten resurssien käyttö tai budjetti. Ne sisältävät mm. ohjelmiston määritykset, laatutavoitteet, aikataulut, kustannukset, tekijänoikeudelliset seikat, ja muut projektiin liittyvät attributit kuten uudelleenkäyttö ja osaamisen kehittäminen. Tavoitteilla tarkoitetaan näistä mitä tahansa, eli tavoite on nimenomaan tarkoituksen, suunnan, tai muun konkreettisen tavoitteen määritelmä.

Tavoitteet on tavallisesti parhaiten määritelty osa ohjelmistoprojektien sopimuksia. Kuitenkin on huomattava, että tavoitteiden asettamisessakin keskitytään usein niiden teknisiin puoliin, kuten ohjelmiston ominaisuuksiin, aikatauluihin ja kustannuksiin. Usein osapuolet asettavat sisäisiä tavoitteita henkilöstön kehityksen, uudelleenkäytön, laadun tai muun suhteen. Nämäkin tavoitteet on syytä kirjata, koska ne vaikuttavat muissa kohdissa tehtäviin sitoumuksiin.

Prosessien määritykset kertovat miten tavoitteet on tarkoitus saavuttaa, miten osapuolten yhteistyö järjestetään, ja miten ohjelmisto kehitetään. Näihin kuuluvat hallinnalliset prosessit, kuten kokouksien järjestelyt pöytäkirjoineen ja itse kehitysprosessin määrittelyt, kuten "käyttöliittymän prototyyppi toimitetaan osana projektin ensimmäistä vaihetta". Prosessien määrittelyt voivat myös viitata ohjelmiston elinkaaren tai prosessien standardeihin, kuten ISO 9000-3 tai SEI Capability Maturity Model.

Usean osapuolen projekteissa prosessien määrittelyt voivat sisältää myös ainakin osittaiset kuvaukset eri osapuolten omista prosesseista ja niiden liittymäkohdista projektiin. Tällä tavalla eri osapuolet voivat ymmärtää paremmin toistensa työskentelytapoja projektin eri vaiheissa.

Riskien ja ongelmatilanteiden hallinta kattaa potentiaalisten ongelmien tunnistamisen ja välttämisen ennakolta, ja määrittelee ennalta mitä tehdään jos ongelmatilanteisiin joudutaan. Ne sisältävät mm. projektin oletukset teknologian mahdollisuuksien, tai ulkoisten tapahtumien vaikutuksen suhteen. Nämä oletukset jäävät usein määrittelemättä, vaikka ne saattavat vaikuttaa merkittävästi projektin osapuolten sitoumuksiin.

Riskien hallinta sisältää osapuolten velvollisuudet riskien suhteen, määritelmän hyväksytyistä riskeistä, ja riskeihin liittyvien kustannusten jaon periaatteet. Ongelmatilanteiden hallinta puolestaan sisältää määrittelyt prosesseista ja ehdoista ongelmatilanteissa, kuten esimerkiksi kompensatioista, rangaistusmaksuista, ja korvausvelvollisuuksista.

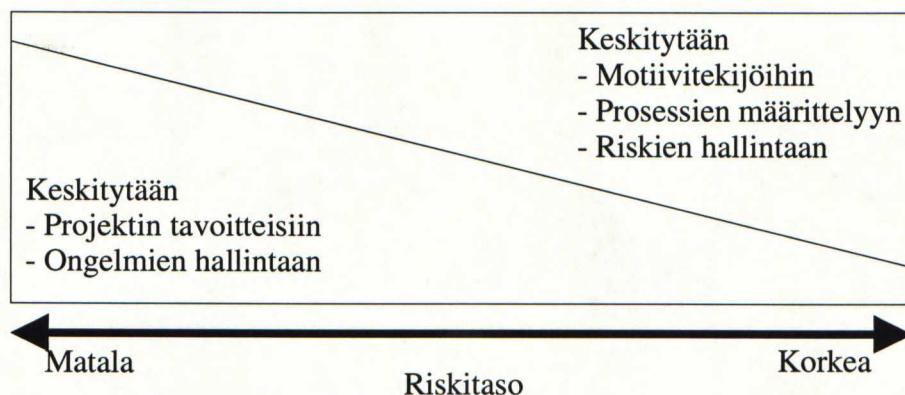
<i>Sitoumusten määrittelyn alue</i>	<i>Aihealueita</i>
Motiivitekijät <i>"Miksi projekti tehdään?"</i>	Liiketoiminnalliset tavoitteet Strategiset suunnat ja aikomukset Yrityksen visio ja tehtävä, ja niiden tulkinta projektissa
Projektin tavoitteet <i>"Mitä toimitetaan ja saavutetaan, milloin ja mihin hintaan?"</i>	Projektin tuotokset Aikataulut Kustannukset Muut tavoitteet ja rajoitteet
Prosessien määrittelyt <i>"Miten projekti tehdään?"</i>	Hallinnalliset menetelmät Kehitysprosessin vaatimukset Muutoshallinnan menetelmät Ristiriitojen ratkaisun menetelmät Prosessin standardit tai puitteet Riskien hallinnan prosessien määrittelyt
Riskien ja ongelmien hallinta <i>"Mitä jos jotain menee vikaan?"</i>	Oletukset - Projektin yhteiset oletukset Riskien hallinta - Riskien hallinnan kattavuus - Hyväksytyt riskit - Riskien vastuiden jakaminen Ongelmien hallinta - Rikkomusten kriteerit - Seuraamukset

Taulukko 1 Sitoumusten hallintamallin sitoumusalueet [Kon98]

Kontion mukaan jokaisessa projektissa ei ole tarkoituksenmukaista määritellä kaikkia alueita perusteellisesti, vaan on syytä keskittyä niihin alueisiin, jotka ovat kyseiselle hankkeelle olennaisimmat ja jotka yleensä voidaan määritellä.

Monet erilaiset tekijät tilanteesta ja projektista riippuen vaikuttavat siihen, mitä sitoumusten alueita tulisi käsitellä ja millä tavalla. Näitä ovat projektin

koko, elinkaaren malli, osapuolten lukumäärä, uusien teknologioiden käyttö jne. Varsinkin projektin alussa tärkein tekijä on kuitenkin koko projektin riskitaso, joten sillä tulisi olla suurin merkitys sitoumuksia määriteltäessä.



Kuva 1 Sitoumusten hallinnan alueet ja projektin riskitaso [ibid]

Kontion mukaan koko projektin riskitason voidaan ajatella olevan projektiin ylipäättään liittyvän arvioitu riski suhteessa muiden vastaavien projektien riskiin. Teoriassa sen voidaan myös ajatella olevan projektin riskeihin liittyvien mahdollisten menetysten summa, mutta tätä harvoin pystytään käytännössä laskemaan.

Korkean riskitason projektissa voi olla turhaa tehdä kiinteitä sitoumuksia projektin tavoitteisiin, koska riskit voivat tehdä näistä tavoitteista mahdottomat saavuttaa. On parempi keskittyä siihen, miten projekti toteutetaan: kokoukset, muutokset hallinnan tavat, raportointi, tiedon vaihto jne. Vastaavasti matalan riskitason projektissa tavoitteisiin voidaan sitoutua huomattavasti enemmän jo alusta asti. Korkean riskitason projekteissa motiivitekijät, prosessien määrittelyt, ja riskien ja ongelmatilanteiden hallinta ovat olennaisempia kuin projektin tavoitteet.

Kontio, Pitkänen ja Sulonen ovat empiirisissä tutkimuksissaan selvittäneet sitoumusten määrittelyä kahdella tavalla. Ensimmäisessä tutkimus tehtiin soveltamalla sitoumusten hallintamallia olemassa olevien ohjelmisto-projektien sopimuksiin. Projektin tavoitteet oli määriteltä kaikissa sopimuksissa, vaikkakin niiden laatu ja tarkkuus oli vaihteleva. Kolmessa sopimuksessa oli käsitelty riskien ja ongelmatilanteiden hallintaa yksityiskohtaisesti, ja muissakin sopimuksissa oli aluetta käsitelty jossain määrin.

Motiivitekijöitä oli käsitelty vain kolmessa projektissa, ja niissäkin vain abstraktisti. Riskien ja ongelmatilanteiden hallinnan yhteydessä oli oletuksia käsitelty vain kahdessa projektissa. Riskien hallinnan prosessit ja riskien tunnistaminen oli käsitelty vain yhdessä projektissa, ja toisessa oli määriteltä joitakin riskien vähentämisen toimenpiteitä. Kaiken kaikkiaan motiivitekijät ja riskien ja ongelmatilanteiden hallinta jäivät useimmin määrittelemättä. Tutkimuksen mukaan riskitaso ei näytä vaikuttavan siihen, miten hyvin näitä alueita käsitellään projekteissa.

Toisessa tutkimuksessa Kontio et al sovelsivat sitoumusten hallintamallia kahteen jo suoritettuun projektiin haastatteleamalla projekteihin osallistuneita

henkilöitä. Molemmissa tapauksissa projektin tavoitteiden tarkempaan määrittelyyn, ja riskien ja ongelmatilanteiden hallintaan arvioitiin kuluvan noin kahdesta kolmeen päivää. Toisessa projekteista tämä myös tehtiin, mikä johti muutoksiin projektissa: projektille määriteltiin uudet viralliset tavoitteet, kohennettu riskien hallinnan menetelmä otettiin käyttöön, ja ongelmatilanteiden käsittelyn prosessia kehitettiin edelleen.

Molemmissa tapauksissa projektihenkilöt päättelivät, että tarkemmat määrittelyt olisi ollut hyödyllistä tehdä tarkemmin projektin alussa, ja päätyivät suosittelemaan että seuraavissa projekteissa noudatetaan sitoumusten määrittelyn prosessia.

Molemmissa tapauksissa projektihenkilöiltä myös tiedusteltiin, miten he olivat valinneet määriteltävät alueet projektin alussa. Alueet oli valittu sen mukaan, mitä projektisopimuksen esimerkeissä tai vakiorungoissa oli määriteltä, ja mitä informaatiota saatiin helposti koottua. Vakiorungot ja informaation saatavuus siis vaikuttavat voimakkaasti sitoumusten määrittelyyn, eikä kumpikaan niistä välttämättä ohjaa määrittelemään juuri niitä sitoumuksia, jotka ovat kunkin projektin kannalta olennaisimpia.

Kontio toteaa tutkimuksen johtopäätöksinä, että sitoumusten hallinnan paras tietämys ja käytäntö riippuu paljon projektihenkilöiden intuitiosta, joka ei useinkaan ole riittävä sitoumusten määrittelyn tärkeys huomioon ottaen. Tekijät päätyvät suosittelemaan, että sitoumusten hallintaan kehitettäisiin käytännöllisiä ohjeita ja menetelmiä.

3.2 Ohjelmistokehityksen prosessimallit

Käsiteltäessä sitoumuksia ohjelmistoprojekteissa laajemmin kuin sopimukseen ja projektisuunnitelmaan kirjoitettuina velvoitteina, niiden ajoittuminen ja taustat määräytyvät projektissa käytettävien prosessien mukaan. Pidemmälle kehittyneissä prosessimalleissa on jopa erikseen määriteltä vaihe päätösten ja jatkositoumusten teolle. Jokaisessa mallissa on joka tapauksessa luonnolliset vaiheet projektin tilan ja jatkon arvioinnille, jolloin myös lähes poikkeuksetta muodostuu osapuolten välille sitoumuksia.

Code-and-fix

Yksinkertaisin ohjelmistokehityksen prosessimalli on ns code-and-fix, jota on käytetty ohjelmistojen kehityksen alkuaikoina pienehköjen sovellusten tuottamiseen. Se pitää sisällään vain kaksi vaihetta: 1) kirjoita koodia 2) testaa koodi ja tee korjaukset. Tässä mallissa vaatimukset, suunnittelu, testaus ja ylläpito tehdään joko kehityksen aikana tai sen jälkeen.

Code-and-fix mallissa on kolme pääasiallista ongelmaa: 1) useiden korjausten jälkeen koodi on epärakenteellista ja jokaisesta korjauksesta tulee vaikeampi. 2) hyvinkin kirjoitettu sovellus ei vastaa käyttäjien tarpeita, ja se usein hylätään tai joudutaan kirjoittamaan uudestaan. 3) koodia on vaikea korjata, koska testausta ja muokkausta ei ole otettu huomioon sitä kirjoitettaessa.

Toteutettaessa suuria ohjelmistoja, kuten SAGE, code-and-fix mallin mukaisesti törmättiin näihin ongelmiin, ja ratkaisuksi kehitettiin vaiheittaisen kehityksen malli. Siinä tunnistetaan ohjelmistokehityksen eri vaiheet: operatiivinen suunnitelma, operatiivinen määrittely, koodauksen

määrittely, koodaus, parametrien testaus, kokoonpanon testaus, rasisustesti ja järjestelmän evaluointi.

Vesiputousmalli

Vaiheittaisesta mallista edelleen kehitetty vesiputousmalli on ollut merkittävä kehitysaskel ohjelmistokehityksen menetelmille. Sen tärkeimmät tarkennukset vaiheittaiseen malliin ovat: 1) vaiheiden välinen takaisinkytkentä, ja ohjeistus rajoittaa palaute edelliseen vaiheeseen, jotta kallis useiden vaiheiden yli annettu palaute minimoidaan. 2) prototyypin ohjelmointi "build it twice"-vaiheen kulkiessa rinnan vaatimusten analyysin ja suunnittelun kanssa. Vesiputousmalli on muodostunut yhdeksi tärkeimmistä ohjelmistokehityksen prosessimalleista. Sitä on laajennettu myöhemmin kattamaan myös asteittaisen kehityksen, rinnakkaisen kehityksen, ohjelmistoperheet, ohjelmistojen evoluution, formaalit menetelmät, sekä vaiheittaisen validoinnin ja riskianalyysin.

Laajennuksista ja tarkennuksista huolimatta vesiputousmalli on useissa projekteissa osoittautunut ongelmalliseksi. Pääasiallinen ongelmien lähde on ollut vaatimus täydellisille dokumenteille jo vaatimuksia analysoinnissa ja suunnittelussa ehtona vaiheiden suoritukselle. Tämä sopii kääntäjien tai käyttöjärjestelmien toteuttamiseen, mutta ei esimerkiksi projekteihin, joissa tavoitteena on interaktiivinen sovellus.

Barry Boehm [Boe88] esittelee myös kaksi muuta mallia: evoluutiomallin, jossa olemassaolevaa ohjelmistoa laajennetaan aina tarpeen mukaan, ja muunnosmallin, jossa muutokset tehdään koodin sijaan aina määrityksiin, ja joka siten soveltuu parhaiten 4GL sovelluskehitykseen. Myös Brooks [Bro95] esittelee yhdistelmiä edellämainituista malleista: inkrementaalinen kehityksen mallin, jossa ohjelmistoon lisätään ominaisuuksia asteittain, ja inkrementaalinen mallin yhdistettynä nopeaan prototyypin luomiseen. Tämän diplomityön puitteissa jätämme näiden mallien käsittelyn, koska ne ovat erikoistapauksia vesiputousmalliin ja seuraavaksi esiteltävään spiraalimalliin verrattuna.

Spiraalimalli

Barry Boehm on esittänyt ohjelmistokehitykseen spiraalimallin, joka on pyrkii ottamaan huomioon nykyisen tavan kehittää laajamittaisia ohjelmistoja. [Boe88] Aikaisemmin mainitut mallit voidaan kuvata spiraalimallilla erikoistapauksina.

Spiraalimallin mukaan sovelluskehitysprosessi jaetaan kierroksiin. Kukin kierros tarkoittaa vaatimusten analyysiä ja projektin suunnitelmaa, sekä lisää projektiin investoituja resursseja edelliseen kierrokseen nähden. Jokainen kierros koostuu samanlaisista vaiheista, ja jokaisen päätteeksi päätetään projektin jatkosta ja sitoutumisesta jatkosuunnitelmiin.

Kierros alkaa vaiheen tavoitteiden määrittelyllä (suorituskyky, toiminnallisuus, jne), toteutusvaihtoehtojen kartoituksella, ja rajoitteiden (kustannukset, aikataulu, jne.) tunnistamisella. Seuraavaksi arvioidaan toteutusvaihtoehtojen suhde tavoitteisiin ja rajoitteisiin. Tässä vaiheessa tunnistetaan yleensä myös epävarmat alueet, joista voi muodostua

projektille merkittäviä riskejä. Jos tällaisia alueita tunnistetaan, niihin liittyvät riskit pyritään minimoimaan kustannustehokkaasti.

Jäljelle jääneet riskit määräävät seuraavan vaiheen. Jos suorituskykyyn tai käyttöliittymään liittyvät riskit ovat kriittisempiä kuin kehityksen ja sisäisten liittymien hallinnan riskit, seuraavaksi saatetaan panostaa tarkempaan suunnitteluun ja seuraavan tason tarkemman prototyypin luomiseen, jotta tärkeimmät riskitekijät pystytään helpommin käsittelemään. Jos näin luotu prototyyppi on käyttökelpoinen jatkokehityksen kannalta, sitä lähdetään seuraavaksi kehittämään edelleen evoluutiomallin tapaan. Toisaalta jos suorituskykyyn ja käyttöliittymään liittyvät riskit on jo ratkaistu, ja kehityksen ja sisäisten liittymien riskit ovat kriittisempiä, seuraavat askeleet ovat vesiputousmallin mukaiset sopivasti muokattuna vaiheittaisen kehityksen tarpeisiin.

Riskien käsittely ohjaa vesiputousmallin vaiheita. Spiraalimalli soveltuu siksi erilaisiin projekteihin, riippumatta siitä painotetaanko niissä määrityksiä, prototyyppejä, simulointia, automaattista generointia tai jotain muuta lähestymistapaa. Lisäksi riskien hallinta määrää miten paljon aikaa ja resursseja panostetaan kuhunkin projektin vaiheeseen, kuten suunnitteluun, konfiguraatioiden hallintaan, laadun varmistukseen, muodolliseen verifiointiin tai testaukseen.

Spiraalimallin suurin etu on juuri sen riskikeskeisyys. Siinä yhdistyvät myös monien muiden mallien hyvät puolet. Hankkeissa, joissa ei ole merkittäviä suorituskykyyn tai käyttöliittymään liittyviä riskejä, se muistuttaa käytännössä vesiputousmallia, ja esimerkiksi automaattista ohjelmiston generointia voidaan käyttää prototyyppien luomiseen. Lisäksi se sisältää luontevan ajankohdan valmiskomponenttien evaluointiin riittävän varhaisessa vaiheessa, ohjaa valmistautumaan ohjelmiston laadun, elinkaaren ja ylläpidon hallintaan, keskittyy huonojen vaihtoehtojen karsimiseen ajoissa, ja sisältää päätöksenteon ja sitoumusten kannalta tärkeät ajankohdat.

Spiraalimallin suurin puute seuraa käytännössä sen riskikeskeisyydestä; projektin onnistuminen on hyvin suuresti kiinni siitä, miten hyvin riskit eri vaiheissa osataan arvioida. Projektissa on siis oltava mukana riittävästi henkilöitä, joilla on ammattitaitoa arvioida riskit oikein, koska parhaatkin riskien arvioinnin menetelmät ovat lähinnä työkalu osaaville henkilöille.

Unified Process, Rational UP

Spiraalimallin pohjalta kehitetty Unified Process on saanut viime vuosina runsaasti huomiota varsinkin oliosuuntautuneen ohjelmistokehityksen prosessimallina. Se on vastaavasti kuin spiraalimalli jaettu vaiheittaisiin kierroksiin, jotka on edelleen jaettu vaiheisiin; Inception, Elaboration, Construction ja Transition. Nämä vaiheet voivat koostua yhdestä tai useammasta iteraatiokierroksesta. Jokainen tällainen iteraatiokierros käsitellään omana miniprojektinaan. Lisäksi malli sisältää joko kokonaan tai osittain rinnakkain tapahtuvat työn vuot, joita ovat mm. vaatimusten hallinta, projektin hallinta, analysointi, suunnittelu, soveltaminen ja testaus. [Jac99]

Unified Processin keskeinen käsite ovat Use Case- määritykset, joilla kuvataan ohjelmiston toiminnallisuutta, joka tuottaa käyttäjälle tai toiselle

ohjelmiston osalle jonkin hyödyllisen tuloksen. Näiden pohjalta suunnitellaan järjestelmän arkkitehtuuri, joka on Unified Process- mallissa keskeisellä sijalla. Use Case- määritykset ohjaavat kaikkia prosessin osia suunnittelusta implementointiin ja testaukseen, ja niillä avulla rakennetulla ohjelmiston mallilla on riippuvuussuhde jokaisen projektin työnvuon malliin. Use Case- määritykset ja muu ohjelmiston kuvaus tehdään Unified Modeling Language- (UML) kielellä.

Muut mallit: XP, Crystal

Extreme Programming- prosessimalli on kehitetty mahdollisimman kevyeksi vaihtoehdoksi nykyisille suhteellisen raskaille kehitysmalleille. Se on erityisesti tarkoitettu ohjelmistoprojekteille, joiden vaatimukset ovat epämääräiset ja muuttuvat jatkuvasti, jotka sisältävät runsaasti riskejä, joilla on tiukat vaatimukset aikataulun ja kustannusten suhteen, ja joissa ei ole mahdollisuuksia panostaa perusteelliseen arkkitehtuurisuunnitteluun ja dokumentointiin. [Bec00]

XP asettaa projektiin vaikuttavat ihmiset ovat keskeiselle sijalle. Sen pääasiallinen motiivi on mahdollistaa näiden ihmisten välinen kommunikointi tunnistamalla keskustelun osapuolten roolit ja määrittelemällä mitä jokaisen osapuolen on velvollisuus kommunikoida toisilleen. [Mar00]

Keskeisimmät XP:n seitsemästä roolista ovat asiakas ja ohjelmoija. Asiakkaan velvollisuus on tunnistaa ominaisuudet (XP:n terminologiassa kertomukset), jotka ohjelmoijan on toteutettava. Samalla asiakas myös määrittelee niiden yksityiskohtaiset hyväksymistestit ja tärkeysjärjestyksen. Ohjelmoijan velvollisuus on toteuttaa asiakkaan haluamat kertomukset asiakkaan määräämässä järjestyksessä, ja läpäisten asiakkaan määräämät testit. Ohjelmoija ei saa toteuttaa mitään, mitä asiakas ei ole erityisesti määritellyt. Ohjelmoijan on toisaalta arvioitava miten kauan toteuttamiseen kuluu aikaa, ja asiakkaan on hyväksyttävä ohjelmoijan arvio.

Tarkkaan määritellyn dialogin avulla halutaan luoda asiakkaan ja ohjelmoijan välille jatkuva keskustelusuhde. Sen avulla ohjelmoija voi saada asiakkaan paremmin luottamaan työmääräarviointihinsa, ja toisaalta siihen, että ohjelmoija todella toteuttaa hänen haluamansa asiat halutussa järjestyksessä.

Toinen XP:n keskeisimmistä käsitteistä on enintään kahden viikon suunnittelujakso. Aika joka kuluu asiakkaan kertomuksen ja sen toteuttamisen välillä on tyypillisesti kaksi viikkoa. Jos toteutus on asiakkaan kannalta väärä, on menetetty vain enintään kahden viikon panos. Tästä seuraa, että XP:ssä suunnittelujakso on enintään kaksi viikkoa, eikä käytännössä mitään yksityiskohtaisia projektisuunnitelmia, analyysimalleja, suunnittelumalleja tai muita tuotoksia tehdä sen pidemmälle. XP:n mukaan pidemmälle meneviä suunnitelmia ei tarvita, koska tiivis tuotosten ja palautteiden sykli poistaa virheet, ja huolehtii siitä että ohjelmisto kehittyy kokonaisuutena oikeaan suuntaan. Asiakkaalle myös kerääntyy kokemusta ohjelmoijan työtahdistista, ja sen perusteella hän voi arvioida, voiko projekti tuottaa tarvittavat ohjelmistot ajoissa. Lisäksi hän voi keskeyttää projektin lähes koska tahansa ja saada silti toimivan kokonaisuuden.

Vaikka XP muistuttaa monessa mielessä Code-and-fix mallia, se pyrkii eroamaan siitä tietyillä laadun varmistamiseen tähtäävillä keinoilla. Ensinnäkin projektissa tuotetun ohjelmakoodin on oltava aina mahdollisimman yksinkertaista, silloinkin kun siihen lisätään ominaisuuksia. Toiseksi ohjelmoijien kuuluu aina työskennellä yhdessä pareittain, jonka ansiosta tuotettu ohjelmakoodi on aina kahden ohjelmoijan välisen keskustelun tulos. Kolmanneksi ennen kuin ohjelmoija lisää koodia järjestelmään, hänen on luotava testi, jota vanha ohjelmisto ei läpäise, mutta muokattu läpäisee.

Sitoumukset eri malleissa

Code-and-fix mallissa puhtaimmillaan päätökset ja sitoutumiset tapahtuvat projektin alussa ja koodauskierrosten välissä. Malli ei määrää tiettyä vaihetta erilaisille sitoumuksille, vaan ne vaihtelevat projektin vaiheen tarpeesta riippuen.

Vesiputousmalli pitää sisällään ajankohdat päätöksille ja sitoumuksille jokaisen putouksen laatikon lopuksi validoinnin ja verifioinnin yhteydessä. Sitoumukset myös vaihtelevat laatikoittain, esimerkiksi projektin resurssit allokoidaan varhaisessa vaiheessa.

Päätösten teko on määritelty spiraalimalliin tärkeänä vaiheena jokaisen kierroksen lopussa. Samassa tilanteessa osapuolet myös sitoutuvat projektin jatkovaiheisiin. Boehm käyttää esimerkkinä spiraalimallin pohjalta tehtyä TRW:n Software Productivity System- kehikkoa, jossa esimerkiksi ensimmäisen kierroksen (feasibility study) päätteeksi sitoudutaan huomattavasti laajempaan resurssien käyttöön (12 mtkk vrt ensimmäisen kierroksen 2). Toisen kierroksen (Concept of Operations) päätteeksi sitoudutaan mm. 100 henkilön käyttöttestaukseen, ja ohjausryhmän perustamiseen hankkeen kokonaisuuden hallintaa varten.

Malli ottaa huomioon tilanteet, joissa jatkokehitystä jaetaan pienemmille tiimeille tai yksittäisille kehittäjille, koska kullekin rinnakkain tapahtuvalle kehitysprosessille voidaan määritellä oma spiraalinsa. Kuitenkin eräs spiraalimalliin heikkouksia on sen hankala soveltuvuus tilanteeseen, jossa ohjelmistoja tilataan ulkoiselta alihankkijalta. Sisäisessä sovelluskehityksessä on suurempi joustavuus vaiheittaisiin, viivästettyihin sitoumuksiin, työpanoksen skaalaamiseen, prototyyppien tekemiseen ja kustannuskeskeiseen suunnitteluun kuin ulkoiselta toimittajalta tilatussa projektissa. Spiraalimallin mukaista ohjelmistoprojektia varten on sopimusten oltava erityisen joustavia. Hankkeen lopputulosta ei tarkkaan tunneta etukäteen, joten osapuolten täytyy tehdä eri tasoisia sopimuksia eri vaiheisiin, kuten puitesopimus, jossa määritellään projektin yleiset suuntaviivat, tavoitteet ja hinnoitteluperiaatteet, sekä erilliset sopimukset projektin vaiheista alkaen tarpeiden kartoituksesta.

Unified Process on pitkälle spiraalimallin kaltainen sitoumusten kannalta. Siinäkin pyritään tuomaan riskit esille mahdollisimman aikaisessa vaiheessa, ja sitoumukset tyypillisesti kehittyvät projektin edetessä. Yhteinen piirre spiraalimallissa ja Unified Process- mallissa vesiputousmalliin verrattuna on myös että projektin hallinta on monimutkaisempaa ja vaativampaa, koska mallit on rakennettu vastaamaan enemmän ohjelmistokehittäjien kuin projektipäälliköiden työtapoja. [Kru00]

Unified Process- malli painottaa myös iteraatioiden hallintaa. Tällä tarkoitetaan aikataulun, resurssien käytön, kulujen, ja teknisten ratkaisujen läpikäyntiä kunkin iteraation päätteeksi, jolloin myös päätetään seuraavasta iteraatiosta. Iteraatioiden hallinnalla pyritään rajoittamaan varsinkin aikataulu- ja kuluriskit mahdollisimman hallittaviin yksikköihin, varmistamaan kehityksen eteneminen, ja kehittämään vaatimuksia vaiheittain.

Sitoumusten kannalta tärkein ajankohta on kunkin iteraation Elaboration-vaiheen lopussa, LCA- vaiheessa. Tässä vaiheessa suurin osa Use Case-määrittelyistä on tehty ja arkkitehtuuri on suunniteltu. Tässä vaiheessa projektipäälliköllä on riittävästi tietoa päättääkseen projektin jatkosta ja resurssien allokoinnista. Tässä ajankohdassa arvioidaan Use Case-määrittelysten ja arkkitehtuurin valmius, ja punnitaan riskit, ennen kuin tehdään päätös sitoutumisesta varsinaiseen kehitysojektiin.

Unified Process- mallissa suositellaan projektisopimuksen teko ainakin kahdessa vaiheessa: ensimmäinen sopimus vain Inception-vaiheesta, ja mahdollinen kiinteähintainen sopimus ohjelmiston toteuttamisesta sen LCA-vaiheessa, eli Elaboration-vaiheen lopuksi.[Kru00]

XP-malli on myös sitoumusten kannalta omintakeinen johtuen sen kahden viikon suunnittelu- ja toteutusjaksosta. Koska suunnittelua ei koskaan tehdä kahta viikkoa pidemmälle, osapuolten sitoumukset ovat myös yhtä lyhytkestoisia. Sekä asiakas että ohjelmoijat voivat suhteellisen helposti neuvotella ja sopia seuraavan kahden viikon sitoumukset. Lisäksi XP auttaa selkeiden roolien kautta tunnistamaan millaisia sitoumuksia osapuolten täytyy tehdä eri tilanteissa.

Toisaalta XP:n lyhyt suunnittelujakso haittaa koko projektin kattavien sitoumusten, kuten motiivitekijöiden, määrittämistä varsinkin suurissa projekteissa. XP tosin sopii muidenkin ominaisuuksiensa ansiosta parhaiten suhteellisen pieniin ja nopeatempoisiin projekteihin, joissa ei ole yli 10 hengen kehitysryhmiä.[Smi00]

3.3 Projektin neuvottelut

Ohjelmistoprojektin aikana käydään monenlaisia neuvotteluja monien eri osapuolten kesken. Ennen kuin sopimus on tehty ja projektin työt aloitettu, osapuolet ovat neuvotelleet hankkeeseen liittyvistä kaupallisista ja sisällöllisistä seikoista, projektin organisaatiosta, ja sen tavoitteista. Projektin aikana neuvotellaan seuraavista vaiheista tai toteutuksen aikana esiin tulleista asioista. Näissä neuvotteluissa päätetään useimmat projektiin liittyvistä sitoumuksista sekä asiakkaan että toimittajan kannalta.

Sopimus- ja muut neuvottelut ovat laajasti tutkittu aihe. Kuitenkin erityisesti ohjelmistoprojekteihin liittyvistä neuvotteluista on tehty suhteellisen vähän tutkimusta. Barry Boehm on kehittänyt ohjelmistoprojektin hallintaan Theory W- mallin [Boe89], joka sisältää WinWin mallin ohjelmistoprojektien neuvotteluille. Theory W:n ydinajatus on löytää projektille sellaiset tilanteet, joissa jokainen osapuoli on voittaja. Boehmin mukaan tämä on jopa projektipäällikön tärkein tehtävä ja projektin onnistuminen riippuu siitä, miten hyvin hän saavuttaa tämän tavoitteen. Boehm on kehittänyt WinWin- mallin pääasiassa projektiin osallistuvien henkilöiden

ohjaamiseen ja motivointiin, mutta sen soveltamisen painopiste voi myös olla organisaatioiden väliset tilanteet.

Theory W sisältää taulukossa 2 esitetyt vaiheet projektin Win-win ennakkoehtojen laatimiseksi, sekä ohjelmistotuotannon prosessin ja ohjelmistotuotteen laatimiseksi.

1. Aseta winwin ennakkoehdot, joilla kaikki voittavat
a. Ymmärrä miten osapuolet haluavat voittaa b. Aseta järkevät odotukset c. Linjaa osapuolten tavoitteet voittamisen ehtojen kanssa d. Varmista tukea tarjoava toimintaympäristö
2. Laadi ohjelmistotuotannon prosessi, jossa kaikki voittavat
a. Laadi realistinen prosessisuunnitelma b. Käytä suunnitelmaa projektin ohjaamiseen c. Tunnista ja hallitse tilanteet, joissa jotkin tai kaikki osapuolet häviävät d. Pidä osapuolet mukana prosessissa
3. Laadi ohjelmistotuote, jonka avulla kaikki voittavat
a. Linjaa tuote käyttäjien ja ylläpitäjien voittamisen ehtojen kanssa

Taulukko 2: WinWin prosessin vaiheet

Jotta ymmärrettäisiin, miten osapuolet haluavat voittaa projektissa, on noudatettava kolmea periaatetta: *On varmistuttava, että avainhenkilöt on tunnistettu.* Usein ohjelmistoprojektit epäonnistuvat, koska kaikkia tärkeitä henkilöitä ei ole otettu mukaan winwin prosessiin. *On pyrittävä asettamaan itsensä toisen osapuolen winwin tilanteeseen.* Oma winwin tilanne ei välttämättä ole winwin tilanne toiselle osapuolelle. *On pyrittävä lähelle asiakasta/käyttäjää.* Ohjelmiston kehittäjiä on suoritettava ohjelmiston suunnittelun ja koodaamisen lisäksi asiakkaita lähellä olevia toimenpiteitä kuten haastatteluja, kyselyjä, operaatioanalyysiä, ja käyttäjien kulttuurin analyysiä.

Monet ohjelmistoprojektit epäonnistuvat, koska asiakkaitten ja käyttäjien odotukset eivät ole järkevällä tasolla. Tämä on seurausta siitä, että käyttäjillä ei ole tuntumaa siihen, mikä on vaativaa ja aikaa vievää, ja mikä suoraviivaista ja nopeaa toteuttaa. Tämän vaiheen tärkeimmät periaatteet ovat:

- Tunnista ja ratkaise kaikki odotusten ristiriidat yhdessä kaikkien osapuolten kesken.
- Varmista että osapuolet tarkastelevat asioita myös toisten osapuolten kannalta.
- Varmista että osapuolet käyttävät objektiivisia, keskenään olennaisia ratkaisukriteereitä

- Suhteuta osapuolten odotukset kokemuksiin: testauksiin, asiantuntijoiden lausuntoihin, vertailuihin vastaaviin projekteihin.
- Suhteuta osapuolten odotukset hyvin perusteluihin malleihin: suorituskyvyn malleihin, ohjelmistoprojektin kustannusten ja aikataulun arvioinnin malleihin.

Jotta osapuolten tavoitteet saadaan linjattua heidän winwin ehtojensa kanssa, on winwin tilanteita aktiivisesti etsittävä ja ratkaisuvaihtoehtoja laajennettava tarvittaessa winwin tilanteiden luomiseksi. Winwin tilanteita voidaan löytää esimerkiksi hajottamalla kokonaisuus osiin (toiminnot, aktiviteetit, kehityssakeet tai vaiheet), joille löydetään sopivat winwin tilanteet. Toinen tekniikka on vaihtoehtojen hakeminen eri osapuolten winwin tilanteista, jolloin esimerkiksi ohjelmiston ylläpidon kehitysvastuu annetaan alusta asti sille osapuolelle, joka tulee toteuttamaan ylläpidon. Ratkaisuvaihtoehtoja voidaan laajentaa yhdistämällä tavoitteita tulevaisuuden ratkaisuihin, laajentamalla tehtävien vastuualuetta, yhdistämällä tavoitteet erityisiin palkkioihin, tarjoamalla tavallista parempaa tukea tehtäville, tai kehittämällä kokonaan uusia ratkaisuvaihtoehtoja.

Theory W- mallin keskeisenä teesinä on, että projekti onnistuu vain jos jokainen projektin osapuoli voittaa. Tätä ydinteesiä tukemaan Boehm on kehittänyt muut Theory W:n periaatteet, kuten riskien tunnistamisen ja hallinnan tärkeyden, sekä projektisuunnitelman laatimisen ja sen noudattamisen tärkeyden. Theory W onkin suunnattu nimenomaan projektin osapuolten hallintaan, ja määrittelee projektin kannalta olennaiset asiat sen kautta, mikä on olennaista siihen osallistuville osapuolille.

Vaikka Theory W ei sellaisenaan sisällä sitoumusten hallintamallin mukaisia ohjeita sitoumusten määrittelyyn, voidaan siitä löytää useita viittauksia samoihin aihealueisiin, jotka sitoumusten hallintamalli määrittelee. Esimerkiksi motivaatiotekijöistä merkittävimmät saadaan todennäköisesti selville, kun selvitetään, miten eri osapuolet haluavat voittaa projektissa. Voittamisen ennakkoehto ovat juuri tärkeimmät syyt osapuolten osallistumiselle projektiin. Samoin järkevien odotusten asettaminen auttaa osapuolia ymmärtämään toistensa näkökulmat ja voiton ennakkoehdot, ja sitä kautta motiivitekijät saadaan määriteltä paremmin.

Toisessa WinWin vaiheessa, jossa perustetaan ohjelmistoprosessi, pureudutaan kolmeen muuhun sitoumusten hallintamallin alueeseen. Realistinen projektisuunnitelma ja projektin ohjaaminen tällä suunnitelmalla sisältävät monia prosessien määrittelyjen ja osittain projektin tavoitteiden aiheita. Win-lose ja lose-lose tilanteiden riskien tunnistaminen ja hallinta kattaa riskien ja ongelmatilanteiden hallinnan alueen suhteellisen hyvin. Osapuolten pitäminen mukana prosessissa puolestaan edellyttää, että projektin prosessimääritykset on tehty riittävällä tarkkuudella.

Boehmin esittämistä esimerkkihankkeista käy ilmi, että monet niissä kohdatuista ongelmista olisi todennäköisesti vältettävissä, jos niiden määrittelyssä olisi käytetty sitoumusten hallintamallia ohjenuorana. Varsinkin projektin motiivitekijöiden määrittely ja niiden vaikutusten ottaminen huomioon olisi todennäköisesti antanut projekteille paremmat edellytykset niitä käynnistettäessä. Boehmin esimerkit ovat erityisen mielenkiintoisia sitoumusten hallintamallin kannalta toisaalta siksi, että ne sisältävät projektille "näkömättömiä" osapuolia, jotka kuitenkin vaikuttavat

ratkaisevasti projektin alkuasetelmiin, ja toisaalta koska joidenkin motiivitekijöiden vaikutus varsinkin projektin aikatauluihin oli negatiivinen, ts projektille asetettiin tiivis ja jyrkkä aikataulu, koska sen alkutilanteeseen vaikutti voimakkaasti projektin kannalta ulkoinen motiivitekijä.

Boehmin TheoryW ja WinWin- malli ovat hyviä menetelmiä sitoumusten määrittelyyn, mutta ne eivät erityisesti ohjaa minkään ennalta määriteltyjen sitoumusalueiden käsittelyyn. Monet sitoumusten hallintamallissa määritelty alueet tulevat käsiteltyä WinWin pohjalta, mutta sitoumusalueet on kuitenkin syytä kartoittaa myös erikseen, jotta olennaiset alueet tulisi käsiteltyä.

Vaatimusten neuvottelut

Eräs ohjelmistoprojektin vaativimmista neuvoteltavista alueista on ohjelmiston vaatimusten neuvottelu. Koko projektin onnistuminen tai epäonnistuminen riippuu usein onnistumisesta vaatimusten määrittelyissä. Monet epäonnistumiset, viivästykset ja budjetin ylitykset voidaan jäljittää puutteisiin vaatimusten määrittelyssä. Vaatimusten joukko ei ole ennalta määrätty, koska eri osapuolet osallistuvat projektiin eri odotuksin ja tavoittein. Toimittaja pääsee perehtymään asiakkaan ja käyttäjän maailmaan, kun taas asiakas ja käyttäjä oppivat lisää siitä, mikä on teknisesti mahdollista ja tehtävissä. Vaatimuksia on usein neuvoteltava osapuolten kanssa, jotka eivät ole varmoja omista tarpeistaan, ja vielä vähemmän muiden osapuolten tarpeista. Vaatimusten neuvottelun on perustuttava osapuolten yhteistyöhön ja aktiiviseen osallistumiseen päätöksenteossa, jotta saavutetaan kaikkia osapuolia tyydyttävä sopimus.

Boehm on kehittänyt WinWin mallistaan EasyWinWin- mallin [Grü01] erityisesti vaatimusten neuvotteluja varten. Sen tarkoituksena on kohentaa keskeisten osapuolten osallistumista ja kanssakäymistä vaatimusten suunnittelussa. Se määrittelee joukon toimenpiteitä jotka ohjaavat osapuolia vaatimusten kokoamisessa, tarkentamisessa, priorisoinnissa ja neuvottelussa. Nämä toimenpiteet ovat:

Neuvoteltavien aiheiden läpikäynti ja tarkennus. Osapuolet yhdessä tarkentavat ja sovittavat neuvoteltavien aiheiden hahmotelman sovellusalueen vaatimusten taksonomian pohjalta.

Osapuolten tavoitteiden brainstorm. Osapuolet kertovat tavoitteensa, näkökulmansa, näkemyksensä, odotuksensa keräämällä lausunnot kunkin menestysehdoista.

Menestyksen ehtojen yhdistäminen. Tämän toimenpiteen tarkoitus on yhdessä luoda lista selkeästi ilmaistuja menestysehtoja edellisen toimenpiteen pohjalta.

Termien määrittely. Osapuolet määrittelevät yhdessä projektin tai aihealueen avaintermit.

Menestysehtojen priorisointi. Osapuolet antavat menestysehdoille tärkeysjärjestyksen, jotta työn kattavuus saadaan määriteltyä ja keskittyminen paranee.

Mahdollisten ongelmien ja rajoitteiden paljastaminen. Mahdollisen ongelmat paljastetaan ja hahmotetaan analysoimalla menestysehtojen tärkeysjärjestys.

Mahdollisten ongelmien, vaihtoehtojen ja sovittavien asioiden tunnistaminen. Osapuolet tunnistavat rajoitteet ja ristiriitaiset menestysehdot mahdollisina ongelmina ja laativat vaihtoehdot niiden ratkaisemiseksi. Näin tehdyt päätökset toimivat perustana sopimusneuvotteluille.

Boehm on selvittänyt WinWin mallin soveltuvuutta ohjelmiston vaatimusten neuvotteluille empiirisellä tutkimuksella [Boe98].

Tutkimuksen mukaan osapuolten menestyksen ehdot eivät olleet ristiriidassa muiden osapuolten ehtojen kanssa. Lisäksi useimmat haasteet olivat erillisiä muista haasteista ja yksinkertaisia ratkaista. Tämän perusteella vaatimusten neuvottelujen menetelmien tulisi keskittyä yhtä paljon yksinkertaisten kuin monimutkaisten suhteiden hallintaan.

Vaatimusten neuvotteluihin käytetty aika ei korreloinut positiivisesti tulosten laadun kanssa. Ryhmät, jotka neuvottelivat vaatimuksensa parissa päivässä saavuttivat parempia tuloksia kuin ryhmät, jotka neuvottelivat selkeästi pidempään. Toisaalta nopeissa neuvotteluissa saavutettiin hyvä laatu vain jos osallistujat olivat riittävän kokeneita.

Tutkimuksen osallistujat olivat IT teknologian asiantuntijoita. Heidän asiantuntemuksensa vaikutti ainoastaan neuvoteltaessa kehitysprosessista ja ylläpidosta, jolloin saavutettiin 30% ajan säästö.

Projekteilla oli tavoitteena tuottaa LCO (Life Cycle Objective) dokumentointi 4 viikon kuluttua vaatimusten neuvottelujen päätyttyä. Näiden dokumenttien laatu voitiin ennustaa ryhmän kokemuksen, neuvottelujen tarkennuskierrosten ja tulosten tuottamisen tehokkuuden perusteella.

Eri osapuolet olivat aktiivisia eri osissa neuvotteluja. Käyttäjät ja asiakkaat olivat aktiivisimpia alkuvaiheessa, jolloin määriteltiin menestyksen ehtoja, kehittäjät ja asiakkaat puolestaan loppuvaiheissa, jolloin määriteltiin sopimukset.

Jotkin ryhmät omaksuivat vesiputousmallin mukaisen neuvottelutavan, kun taas toiset ryhmät tarkentavan spiraalimallin tapaisen tavan. Jälkimmäiset ryhmät saivat aikaan LCO kriteereillä mitattuna laadukkaamman dokumentaation kuin ensin mainitut.

Kokeneemmat ryhmät saivat aikaan laadukkaammat LCO dokumentit vähemmällä vaivalla kuin vähemmän kokeneet ryhmät. Jotkin vähemmän kokeneet ryhmät saivat myös aikaan laadukkaan LCO dokumentaation, mutta joutuivat käyttämään siihen runsaasti aikaa ja vaivaa.

Winwin menetelmää käytettiin noin neljännes neuvotteluihin käytetystä ajasta. Menetelmä vaikutti positiivisimmin yhteistyöhön, avainasioihin keskittymiseen, kitkatekijöiden vähentämiseen, ja hajautetun yhteistyön mahdollistamiseen. Lisäksi menetelmä teki osapuolista tasaveroisemmat.

Vaatimusten neuvotteluissa tehdään monet tärkeimmistä sitoumuksista, jotka ovat varsinkin sitoumusten hallintamallin Projektin tavoitteet-aihealueella. Boehmin EasyWinWin- malli näyttää tutkimusten perustella antavan hyvät työkalut myös tämän alueen neuvotteluihin. Mallin

pääasiallinen tarkoitus on kuitenkin neuvottelujen tehostaminen ja lopputulosten kehittäminen tämän kautta, eikä se painota erityisesti sitä, mistä aihealueista tulisi neuvotella.

3.4 Ohjelmistosopimukset

Juhani Warsta on väitöskirjassaan tutkinut ohjelmistoprojekteissa tehtäviä sopimuksia varsinkin liiketaloudellisesta näkökulmasta [War01]. Hänen tutkimuksensa kohteena ovat olleet erityisesti sellaiset suomalaiset ohjelmistoyritykset, jotka harjoittavat myös vientitoimintaa ja joilla on sen takia tarve tehdä kansainvälisiä sopimuksia. Warsta tutki erikseen vakio-ohjelmistoja, muokattavia ohjelmistoja ja asiakaskohtaisia ohjelmistoja tuottavien yritysten sopimukset.

Warstan tutkimuksessa tavoitteena oli selvittää, miten organisaatiot ovat lähestyneet ja mallintaneet ohjelmistosopimusten tekoa, millainen rooli organisaatioiden välisillä ja niiden sisäisillä suhteilla on sopimusten teossa, miten sopimusprosessit syntyvät ja kehittyvät osana ohjelmistoliiketoimintaa, millaisilla malleilla näitä prosesseja ja suhteita voidaan hahmottaa, ja miten sopimusprosesseja voidaan hallita osana ohjelmistoliiketoimintaa. Vaikka Warstan tutkimuksessa ei erityisesti painoteta projektissa tehtäviä sitoumuksia, on joistakin hänen tuloksistaan tulkittavissa erilaisten sopimussuhteiden vaikutus myös sitoumuksiin.

Warstan mukaan ohjelmiston kehityssopimusten ja sopimusneuvottelujen keskeisimmät alueet ovat kohteen määrittelyt, tarpeelliset käyttö-, omistussenssi- ja muut IPR oikeudet, projektin hallinta, ja ohjelmiston tuki ja jatkokehitys. Lisäksi Warstan mukaan hyvän sopimuksen tulisi sisältää tai ainakin osapuolten tulisi ymmärtää ohjelmiston hinnoittelun ehdot ja rakenteen, projektin aikataulun, välimiesmenettelyn, takuut, sekä korvausvastuut ja vastuurajoitukset. Lisäksi Warsta huomauttaa, että koska tärkeitä alueita on näin monta, ja ne ovat vaatimuksiltaan hyvin erilaisia, sopimusten neuvottelu vaatii syvällistä tietoa ja kokemusta monelta alueelta, varsinkin koska ne käydään usein kovien aikataulupaineiden alla.

Warstan tutkimuksissa informaation epäsymmetria tuli esille voimakkaasti vaikuttavana ympäristötekijänä useissa projekteissa. Tällainen tilanne on esimerkiksi silloin, kun suuren asiakkaan neuvottelukumppani on pieni toimittaja, jolla ei ole omaa lakiasiaain osastoa. Tällöin toimittaja on neuvoteltaessa heikommassa asemassa kaikkiin sopimukseen ja sen ehtoihin liittyvissä asioissa. Vastaava tilanne saattaa syntyä silloinkin, kun toimittajalla on asiakkaaseen nähden ylivertainen tekninen tietotaito. Tällainen tilanne voi syntyä, vaikka toimittaja olisi merkittävästi asiakasta pienempi.

IPR oikeudet ovat Warstan tutkimuksen mukaan tärkein ja keskeisin ohjelmistosopimusten alue. Käytännöt IPR oikeuksien suhteen vaihtelevat paljon sekä projekteittain, yrityksittäin, että maittain. Kun Suomessa on tapana, että IPR oikeudet siirtyvät ohjelmiston tilaajalle, niin Yhdysvalloissa ne sen sijaan jäävät toimittajalle, ja asiakas saa ohjelmistoihin käyttö-oikeuden ilman oikeutta siirtää sitä toiselle.

Yhteistyö eri tasoilla on Warstan mukaan myös yksi keskeisimmistä teemoista ohjelmistojen kehityksessä. Tällä Warsta viittaa sekä ohjelmistoyritysten väliseen yhteistyöhön, että yritysten yhteistyömuotoihin

asiakkaittensa kanssa. Yhteistyöllä on voimakas yhteys koko teollisuudenalan nopeaan muutokseen ja kehitykseen.

Ohjelmistoyritysten yhteistyö keskenään ja asiakkaittensa kanssa perustuu siihen, että jokin yritys pystyy tarjoamaan sellaista ydinsaamista, joka toiselta yritykseltä puuttuu ja joka sopii sen strategisiin linjauksiin. Näiden suhteiden muodostumiseen ja varsinkin niiden menestykseen vaikuttavat asiakaan kokemus yhteistyöstä toimittajien kanssa, toimittajan perehtyneisyys asiakkaan tai kumppanin tarpeisiin, ja aikaisemmat projektit.

Lappi on myös tutkinut samaa materiaalia kuin Warsta, mutta suhteiden aloittamisen kannalta [Lap01]. Hän jakoi tarvittavat elementit kolmeen alueeseen: soveltuvat tarpeet ja resurssit, toimijasta riippuvat tekijät, ja tulevaisuuden odotukset. Soveltuvista tarpeista ja resursseista tärkein tekijä oli projektissa tarvittavat resurssit ja valmiudet. Muita tärkeitä tekijöitä olivat asiakkaan tarjonnan valikoima ja toimittajan asema markkinoilla. Toimijasta riippuvista tekijöistä tärkeimmät olivat puolestaan toimittajan koko ja maine, sekä henkilökohtaiset suhteet. Tulevaisuuden odotuksia määrittelevät eniten keskinäiset tavoitteet ja edut, pitkän aikavälin suuntautuminen ja osapuolten sitoutuminen. Tällä alueella tärkein tekijä on pitkän aikavälin taloudellinen etu, mutta yritykset pyrkivät myös pitkän aikavälin suhteisiin ja sitoutumiseen alusta asti.

Warsta on tutkinut erikseen valmisohjelmistojen, räätälöityjen ohjelmistojen ja muokattavien valmisohjelmistojen toimitusprojekteihin liittyviä sopimusprosesseja. Koska tässä työssä käsitellään ohjelmistoprojektien sitoumuksia, kiinnostavimmat hänen tutkimusalueistaan ovat varsinkin räätälöityjen ohjelmistojen, mutta myös muokattavien valmisohjelmistojen sopimusprosessit. Warstan käyttämässä mallissa prosessissa on tunnistettu neljä osaa: neuvotteluvaihe, määrittelyvaihe, tuotantovaihe ja arviointivaihe.

Räätälöityjen ohjelmistojen projektit

Warstan mukaan räätälöityjen ohjelmistojen projekteille on luonteenomaista toimittajan ja asiakkaan läheisempi ja pitkäaikaisempi suhde, joka perustuu molemminpuolisiin sitoumuksiin ja ymmärrykseen. Tämä suhde myös sekä vaatii että siitä seuraa pitkäaikainen molemminpuolinen riippuvuussuhde. Tästä seuraa, että asiakkaan ja toimittajan välinen puitesopimus on keskeisessä asemassa räätälöityjen ohjelmistojen projekteissa, koska siitä neuvoteltaessa luodaan perusta tulevalle yhteistyölle. Warstan mukaan jotkin yrityksistä eivät edes halunneet aloittaa tarkempia projektineuvotteluja ennen kuin puitesopimuksesta oli päästy yhteisymmärrykseen.

Puitesopimus on sitoumusten kannalta tärkeä, koska se tuo jatkuvuutta asiakkaan ja toimittajan suhteeseen. Sen avulla päästään tilanteeseen, jossa uusien projektien käynnistäminen on helppoa, mikä edelleen johtaa siihen, että projekteja on käynnissä jatkuvasti. Tällaisessa tilanteessa asiakas ja toimittaja voivat molemmat sekä suunnitella että sitoutua pitkän aikavälin tavoitteisiin, kuten henkilöstön osaamisen ja muiden valmiuksien linjaamiseen ennakoitujen tarpeiden mukaan. Puitesopimukseen vaikuttavat merkittävästi sitoumusten hallintamallin motiivitekijät.

Kun puitesopimus on saatu valmiiksi, asiakkaan ja toimittajan interaktio voi siirtyä seuraavaan vaiheeseen, jossa neuvotellaan yksittäisistä

ohjelmistoprojekteista. Asiakas laatii projektin tarjouspyynnön, johon toimittaja vastaa projektitarjouksella. Tarjousta useimmiten vielä muokataan tarjousneuvotteluissa. Kun asiakas lähettää toimittajalle muokattuun tarjoukseen ja puitesopimukseen viittaavan työtilauksen projektista, saatetaan joissakin tapauksissa vielä neuvotella erillinen toimitussopimus, joka on tarjousta yksityiskohtaisempi sopimus. Tällainen sopimus tehdään varsinkin sellaisissa hankkeissa, joissa lopullisen tuotoksen vaatimia panostuksia ei pystytä arvioimaan ennen kuin perusteellisempi selvitys tai kokeiluprojekti on suoritettu.

Koska puitesopimuksen avulla yksittäisten projektien neuvottelut ovat yksinkertaisempia, saatetaan joissain yrityksissä käyttää projektitilauksia, kun halutaan aloittaa hanke mahdollisimman nopeasti. Projektitilaus on kevyt, ennalta määritelty sopimus, jolla asiakas voi tilata varsinkin sellaisia hankkeita, joiden kaltaisia toimittaja on aikaisemminkin toimittanut samalle asiakkaalle. Niiden merkitys on sitä suurempi, mitä kehittyneempi asiakkaan ja toimittajan suhde on.

Projekti-sopimuksen neuvottelujen määrittelyvaiheessa sovitaan projektin tavoitteet sellaisella tarkkuudella, että toimittaja pystyy arvioimaan projektin tarpeet riittävällä tarkkuudella, jotta varsinainen ohjelmointityö voitaisiin aloittaa. Asiakkaan ja toimittajan aikaisempi yhteistyö vaikuttaa myös määrittelyvaiheeseen, koska silloin toimittaja ymmärtää paremmin asiakkaan tarpeet, ja asiakas puolestaan toimittajan valmiudet.

Tuotantovaiheessa joudutaan useimmiten tekemään määrittelyihin joko ennakoituja tai ennakoimattomia muutoksia. Warstan mukaan näiden muutosten lähde on olennainen sopimusten kannalta. Jos muutoksia haluaa asiakas, neuvotteluista tulee molemmille osapuolille helpommat, varsinkin jos muutokset aiheuttavat lisäkustannuksia.

Ohjelmiston toimituksen jälkeen projekti etenee arviointivaiheeseen. Tässä vaiheessa suoritetaan toimituksen arviointi, tarvittaessa sopimuksen tarkistaminen, ja palaute projektista ja koko yhteistyöstä. Warstan havaintojen mukaan tässä useat yritykset pitivät hyvänä asiakkaan ja toimittajan suhteen päättämistä erillisellä sopimuksella, vaikka tavallisesti näin ei tehdäkään.

Muokattavien valmisohjelmistojen projektit

Muokattavien valmisohjelmistojen liiketoiminta on sekoitus valmisohjelmistojen ja räätälöityjen ohjelmistojen liiketoimintoja. Määritelmän mukaan muokattavien valmisohjelmistojen toimittaja on kehittänyt ja testannut merkittävän, usein suurimman osan toimitettavasta ohjelmistosta ennen kuin sitä aletaan markkinoimaan. Pienempi osa lopullisesta ohjelmistosta tehdään asiakkaan tarpeiden ja määritysten mukaan. Tämän pienemmän osa tekemiseen sisältyy olennaisesti myös ohjelmiston valmiin osan muokkaaminen, jotta se saadaan sopimaan asiakkaan ympäristöön. Toisin kuin valmisohjelmistojen toimituksissa, joissa sovittaminen tehdään parametroimalla, näissä toimituksissa muutokset tehdään ohjelmiston lähdekoodiin.

Muokattavien valmisohjelmistojen toimittaja pyrkii myös yhdistämään valmisohjelmistojen ja räätälöityjen ohjelmistojen liiketaloudelliset edut. Hän pystyy sopeutumaan muuttuviin vaatimuksiin huomattavasti

nopeammin kuin valmisohjelmistojen toimittaja, eikä kärsi yhtä helposti kilpailevista tuotteista. Hän ei ole toisaalta yhtä resurssirajoitteinen kuin räätälöityjen ohjelmistojen toimittaja, koska hän pystyy hyödyntämään kerran kehitettyä koodia useassa projektissa, ja pystyy siten hyödyntämään laajempia markkinoita.

Warstan mukaan puitesopimuksia käytetään muokattavien valmisohjelmistojen toimituksissa yhtä laajasti kuin valmisohjelmistojen ja räätälöityjen ohjelmistojen tapauksissa. Puitesopimusten painopisteet olivat lähempänä valmisohjelmistojen kuin räätälöityjen ohjelmistojen puitesopimuksia. Sopimusten perusta on ohjelmiston lisensointi samalla tavalla kuin valmisohjelmistojen tapauksessa. Lisäksi keskeisiä alueita ovat teknologian siirron sopiminen, toimitussopimukset jakelijan kanssa, ja projektisopimus loppuasiakkaan kanssa. Projektisopimuksessa sekä määrittellään tarkkaan ohjelmiston liitynnät asiakkaan ympäristöön, että kuvataan toimittajan suorittamat tehtävät projektissa.

Samoin kuin räätälöityjen ohjelmistojen tapauksessa, myös muokattavien valmisohjelmistojen projektit jaetaan erillisiin määrittely- ja toteutusprojekteihin. Määrittelyprojekti on tarpeellinen varsinkin kun asiakkaan tarpeita ei voida tarkkaan määritellä. Usein toimittaja tekee tarjouksen toteutusprojektista vasta kun määrittelyprojekti on saatettu loppuun. Lisäksi kirjoittajan kokemuksen mukaan monet muokattavien valmisohjelmistojen toimittajat ovat kehittäneet toimitettaville ohjelmistoille suunniteltuja menetelmiä vaatimusten selvittämiseen.

Warstan mukaan keskeinen haaste ohjelmistoprojektien sopimusten teossa on sisäisten sitoumusten aikaansaaminen. Tarvittavat sisäiset sitoumukset pyritään kehittämään tehtäviä määriteltäessä, jolloin projektiin osallistuvat toimittajan sisäiset ryhmät pääsevät ilmaisemaan omat arvionsa aikatauluista ja tarvittavista resursseista. Samalla sisäiset ryhmät määrittelevät aikatauluihin joustovarot yllättäviä tarpeita varten.

Muokattavien valmisohjelmistojen luonteen mukaisesti toteutusprojektit voivat olla hyvinkin lyhyitä, Warstan mukaan jopa vain viikon tai kahden mittaisia. Projektin pituuden todettiin myös vaihtelevan merkittävästi riippuen siitä, miten monta kyseisen ohjelmiston toimitusprojektia toimittaja oli suorittanut. Koska ensimmäisessä projektissa toimittajalla ei vielä ole paljoakaan kokemusta ohjelmistosta, projekti saattaa kestää useita kuukausia. Warstan mukaan tarvitaan kolmesta neljään projektia, jotta toimittaja saa riittävästi kokemusta ohjelmiston muokkaamisesta ja toimittamisesta.

Toimitusvaihe saatetaan Warstan mukaan jakaa eri tasoihin toimitusvaiheisiin, jotka tarjotaan erikseen. Ensimmäinen vaihe on tavallisesti toimiva asennus yhdestä ohjelmistotuotteesta tai osasta ohjelmistoperhettä. Kun perusohjelmisto on toimitettu, toimittaja alkaa määritellä seuraavia vaiheita jollain tarkkuudella. Kun vaihe kaksi on toimitettu, vaiheen kolme määrittelyt tarkennetaan, ja näin edetään niin monessa vaiheessa kuin on tarpeellista. Joissakin tapauksissa toimittaja kuvailee koko projektin, tuotteen ja ominaisuudet heti ensimmäisessä tarjouksessa, ja tarjouksessa on joitakin kohtia, joita tarkennetaan projektin edetessä.

Projektin arviointivaihetta tutkiessaan, Warsta tuo esille kantansa ohjelmistoprojektien sopimusprosessin kahdesta tärkeimmästä

tarkoituksesta. Tärkein tarkoitus on osapuolten välisen suhteen ja yhteisen hankkeen kuvaus ja suhteiden suojaaminen yllättävien tapahtumien vaikutuksilta. Toiseksi tärkein tarkoitus on osapuolten tutustuminen toisiinsa ja toistensa tapoihin tehdä projekteja ja liiketoimintaa.

Warstan tutkimuksissa ilmeni vain vähän projekteja, joiden sopimus oli toimittajan kannalta huono. Pääasiassa sopimuksen teki huonoksi se, että toimittaja joutui tekemään projektissa enemmän työtä kuin mistä voitiin laskuttaa asiakasta. Jotkin tällaiset huonot projektit tosin ajateltiin hyödyllisiksi tulevaisuudessa joko jatkohankkeina, oppimisprosesseina tai hyvinä referenssiprojekteina myydessä muille asiakkaille.

Warstan havaintojen mukaan muokattavien valmisohjelmistojen liiketoiminta on kiinnostava yhdistelmä valmisohjelmistojen ja räätälöityjen ohjelmistojen liiketoimintaa. Liiketoiminnassa on elementtejä valmisohjelmistoista, kuten lisensoinnin ja tekijänoikeudellisten seikkojen korostaminen. Lisäksi toimituskanavan, kuten jälleenmyyjien ja integraattoreiden, merkitys on suuri. Myyntiprosessi puolestaan muistutti paljon räätälöityjen ohjelmistojen myyntiä useine tarjouskierroksineen. Samoin toimitusvaihe on hyvin samankaltainen kuin räätälöityjen ohjelmistojen projekteissa.

Asiakkaan ja toimittajan suhteen päättäminen

Warsta on myös tutkinut miten asiakkaan ja toimittajan välinen suhde päättyy. Tavallisimmin suhde vain hiipuu, kun osapuolet lakkaavat pitämästä yhteyttä koska ne eivät enää ole kiinnostuneita toistensa toiminnasta. Tällainen tilanne syntyy esimerkiksi silloin, kun asiakkaan puolella ohjelmistoprojektiin ei ole sitoutunut koko yritys, vaan ainoastaan tietty henkilö, joka on siirtynyt pois yrityksen palvelusta.

Joissakin tapauksissa suhde lopetettiin erityisellä sopimuksella. Syitä lopettamiseen ei tutkimuksessa saatu selvitettyä, mahdollisesti koska lopettaminen johtui ongelmista projektissa.

Suhteen päättäminen oli tutkimuksen mukaan ominaista varsinkin valmisohjelmistojen tapauksessa, pääasiassa koska niiden liiketoiminnan luonne ei kannusta jatkuvaan suhteeseen. Näissäkään tapauksissa suhdetta ei lopetettu erityisellä sopimuksella, vaan kanssakäyminen vain loppui hiljalleen.

Warstan tutkimuksen mukaan päättämistä kiihdyttäviä tekijöitä olivat asiakkaan merkittävät organisaatiomuutokset, asiakkaan tuotestrategian muutokset, asiakkaan oman liiketoiminnan ajautuminen vaikeuksiin, asiakkaan avainhenkilöiden henkilökohtaisten suhteiden vaikutus, ja huono sopimus, jota ei jostain syystä onnistuta neuvottelemaan uudestaan. Tärkeimpiä päättämistä hidastavia tekijöitä olivat puolestaan yhteiset liiketoiminnalliset edut ja ennen kaikkea toimivat henkilösuhteet asiakkaan ja toimittajan projektihenkilöiden välillä. Jälkimmäisen todettiin jopa auttamaan suhdetta kestämaan muita, jopa liiketaloudellisia ongelmia. Henkilösuhteiden todettiin olevan erityisen tärkeitä räätälöityjen ohjelmistojen ja muokattavien valmisohjelmistojen tapauksissa.

3.4.1 Ohjelmistosopimuksen vakiopohjat

Ohjelmistotuotanto on monimutkaista liiketoimintaa, jolle ovat ominaisia nopea kehitys ja vaihtelevat liiketoimintasuhteet. Lisäksi sillä on monia erityispiirteitä; tuotettu hyödyke ei ole käsin kosketeltava, se sisältää usein merkittävän määrän palveluja, ja asiakas pääsee perehtymään tuotteeseen perusteellisesti vasta kun sopimukset ovat allekirjoitettu. Mm. näistä syistä johtuen ohjelmistoteollisuus on kehittänyt vakiosopimuksia helpottamaan sopimusten neuvotteluja. Lisäksi monilla ohjelmistoja tuottavilla yrityksillä on omia vakiosopimuksiaan.

Vakiosopimukset on useimmiten laadittu siten, että ne ottavat huomioon mahdollisimman laajan skaalan erilaisia liiketoiminnan tilanteita. Tästä syystä vakiosopimuksia käytettäessä on useimmiten käytännöllistä valita niistä ne osat, jotka ovat kussakin tapauksessa tärkeimmät. Usein kuitenkin esimerkiksi yhtiön politiikka sanelee, että vakiopohjia on käytettävä mahdollisimman kokonaisina. Vakiosopimukset tavallisesti keskittyvät juridiikan kannalta keskeisimpiin seikkoihin, kuten takuu, käyttöoikeudet, keskeyttämis ehdot, korvaukset jne. Ne eivät tavallisesti puutu projektin organisointiin, aikatauluihin tai ohjelmiston määrittelyihin.

IT2000

IT2000 on Keskuskauppakamarin, Suomen Logistiikkayhdistys ry:n, Tietotekniikan liitto ry:n ja Tietotekniikan palveluliitto TIPAL ry:n yhdessä yrityksille laatimat sopimusehdot ja -mallit toimittajan ja asiakkaan välisille toimituksille. Niiden laatimiseen on myös osallistunut joukko toimittajien ja asiakkaiden edustajia [Kes01].

IT2000 sopimusehdot koostuvat IT2000 YSE yleiset sopimusehdot- liitteen lisäksi erityisehtoliitteistä, joita sovelletaan projektissa toimitettavan kohteen mukaan ja projektin eri vaiheissa. Vakio-ohjelmistojen ja asiakaskohtaisen ohjelmistojen toimituksille on erilliset erityisehtonsa, koska niiden toimituksilla on erilaiset vaatimukset mm. käyttöoikeuksien, muutostentien oikeuksien, vastaanottokriteerien ja takuiden osalta.

Erityisehtoliitteet ovat:

- IT2000 ELT erityisehtoja laitetoimituksista
- IT2000 EVT erityisehtoja valmisohjelmistojen toimituksista
- IT2000 EAT erityisehtoja asiakaskohtaisen ohjelmiston toimituksesta
- IT2000 EAP erityisehtoja konsultointi- ja muista asiantuntijapalveluista
- IT2000 ELH erityisehtoja laitteiden huoltopalveluista
- IT2000 EOY erityisehtoja ohjelmistojen ylläpitopalveluista

Lisäksi IT2000 sisältää IT2000 PTE pientoimitusehdot, joka on yksinkertaistettu versio yleisistä sopimusehdoista tilanteeseen, jossa toimituksen kohteena on ainoastaan vakioita laitteisto- ja ohjelmistotuotteita, ja kertaluonteisia palveluja esimerkiksi tuotteiden asentamiseksi.

Ohjelmistoprojektien sitoumusten kannalta kiinnostavimpia IT2000 ehtoja yleisten sopimusehtojen lisäksi ovat varsinkin erityisehdot asiakaskohtaisen ohjelmistojen toimituksista. Erityisehdot konsultointi- ja asiantuntija-

palveluiden toimituksista saattavat myös joissain tapauksissa olla sopivin liite, varsinkin jos ohjelmistoprojektin sopimuksen painopiste on asiantuntijoiden suoritteessa eikä ohjelmiston toimittamisessa. Tällainen tapaus voi olla esimerkiksi ulkopuolisten asiantuntijoiden hankkiminen oman henkilöstön tueksi, jossa ulkopuoliset asiantuntijat halutaan sitouttaa projektiin muutenkin kuin ostamalla työtunteja.

Yleisissä sopimusehdoissa määritellään ennen kaikkea riskien ja ongelmatilanteiden hallintaan liittyviä alueita, koska vakiosopimus pohjille on tyypillistä juuri poikkeustilanteiden painottaminen. Ehdoissa määritellään mm. menettelyt ja osapuolten vastuut ylivoimaisen esteen, toimituksen viivästymisen, sopimusrikkomusten ja sopimuksen purkamisen tapauksessa.

EAT – erityisehdot asiakaskohtaisten ohjelmistojen toimituksista määrittelee joitakin prosessimäärittelyjen osia, vaikkakin suurin osa ehdoista on suunnattu projektin juridisten velvoitteiden määrittelyyn. Kohta 4: projektin organisointi ja toteutus sisältää varsinkin prosessimäärittelyyn liittyviä asioita, kuten projektin perusorganisaation, työskentelytilat, ja varmuuskopioinnin. Lisäksi kohdat 6: raportointi, 7: toimitus ja asennus sekä 8: testaus ja toimituksen hyväksyminen sisältävät joitakin kohtia prosessimäärittelyyn liittyen. Kohta 9: toimituksen viivästyminen koskettaa puolestaan riskien ja ongelmatilanteiden hallintaa, kuitenkin vain viivästymisen aiheuttamien korvausten osalta.

EAP – erityisehdot konsultointi- ja muista asiantuntijapalveluista määrittelee myös joitakin prosessimäärittelyjen osia. Kohdassa 4: palvelu ja sen tulosten hyväksyminen määritellään joitakin prosessimäärittelyjen osia, kuten raportointi, osien hyväksynnät, tehtäviä suorittavat henkilöt, ja virheiden korjaukset.

Kokonaisuudessaan IT2000 vakioehdot koskettavat varsin ohuesti sitoumusten hallintamallin mukaisia sitoumuksia. Sitoumusalueista määritellään lähinnä prosessimäärittelyihin ja riskien ja ongelmatilanteiden hallintaan liittyviä asioita, mutta nekin käsitellään lähinnä juridisesta näkökulmasta. Ne keskittyvät pääasiassa sopimuksen juridisiin perusmäärittelyihin, mikä on todennäköisesti ollut myös laatijoiden tarkoitus. IT2000 on laadittu nimenomaan sopimusten tekoon eikä niissä puututa mitenkään toimitusprojektin sisältöön, suunnitelmaan tai hallintaan. IT2000 vakioehdot ovat hyödyllisimmillään, kun niitä täydennetään sellaisilla projektisuunnitelmalla, työsuunnitelmalla ja projektin hallinnan menetelmällä, joissa otetaan huomioon projektissa tehtävät sitoumukset.

Sun Microsystems: Master Consultancy Agreement

Tämä vakiosopimus on kehitetty toimimaan vakio pohjana Sun Microsystems:in Asiantuntijapalvelujen projektisopimuksille. Se on vastaava vakiosopimus kuin IT2000, koska sekin keskittyy ennen kaikkea juridisten seikkojen kattamiseen, eikä puutu itse projektin toimittamiseen. Osittain tämä johtuu myös siitä, että Sun Asiantuntijapalvelut toimittavat suhteellisen laajalla skaalalla erilaisia projekteja, ja vakio pohjan on sovellettava näihin kaikkiin. Tästä syystä se ei myöskään sisällä mitään muuta erityisesti ohjelmistoprojektien tarpeisiin suunnattua kuin tekijänoikeudelliset pykälät.

3.1 Ohjelmistoprojektin hallinta

IEEE 1058.1 standardi ohjelmistoprojektien hallinnalle

Kuten aikaisemmin on mainittu, ohjelmistoprojektien projektisuunnitelmille on olemassa IEEE standardi 1058.1 vuodelta 1987 [IEE87]. Tämän standardin on tarkoitus määritellä projektisuunnitelman muoto ja sisältö, mutta se ei määrittele miten kyseiseen suunnitelmaan on päädytty, eikä tarjoa esimerkkejä projektisuunnitelmista. Projektisuunnitelman laatimisen ja ylläpidon menetelmien ja prosessien kehittäminen on jätetty standardin käyttäjien vastuulle. Standardi on tarkoitettu kaiken tyyppisille ohjelmistoprojekteille, riippumatta siitä onko niiden tarkoituksena ohjelmakoodin tuottaminen, ohjelmiston suunnittelu, olemassaolevan ohjelmiston muokkaus, ohjelmistotuotteiden määrittely tai toteuttamiskelpoisuuden esiselvitys (*feasibility study*?). Standardi on myös tarkoitettu eri kokoisille projekteille. Pienemmät projektit eivät vaadi yhtä pitkälle vietyä muodollisuutta kuin suuret, mutta standardin määrittelemät osat pätevät kaikkiin projekteihin.

Standardi määrittelee projektisuunnitelmaan joitakin seikkoja sitoumusten hallintamallin sitoumusalueista.

Projektisuunnitelman esipuhe sisältää yhden tärkeimmistä motiivitekijöistä, projektin tarkoituksen. Standardi ei kuitenkaan kuvaa tarkemmin, miten projektin tarkoitus tulisi määritellä.

Projektin yleiskatsaus (1.1) sisältää projektin tavoitteet- alueen määrittelyjä: projektin tavoitteet, tuotettava ohjelmisto, tärkeimmät työvaiheet ja niiden tuotokset, tärkeimmät virstanpylväät, tarvittavat resurssit, budjetti ja korkean tason aikataulu. Samassa kappaleessa määritellään tarvittaessa myös projektin motiivitekijöihin kuuluva suhde muihin projekteihin.

Projektin tuotokset- kohdassa (1.2) kuvaillaan, mitä projektin on tarkoitus tuottaa asiakkaalle, missä aikataulussa, paikassa, ja millaisina määrinä. Lisäksi kohdassa Viitemateriaali (1.4) kuvataan yksityiskohtaisesti projektin tuottama kirjallinen materiaali. Nämä sisältyvät myös projektin tavoitteet- alueeseen.

Kohdassa Prosessimalli (2.1) kuvataan projektin päätoimintojen ja aktiviteettien välinen suhteet tärkeimpien virstanpylväiden, katsauksien, tuotosten, toimitettavan materiaalin, ja kuittausten ajoituksen koko projektille. Prosessien määrittäminen- kohdan sitoumukset määritellään pääosin tässä kohdassa. Niitä määritellään lisäksi kohdissa 2.2-2.4, joissa kuvataan projektin organisaatorakenne, organisaatioiden rajat ja liittymät, sekä projektin vastuut.

Projektin tavoitteita, prosesseja ja osittain motiivitekijöitä määritellään myös kohdissa 3.1 ja 3.2, joissa kuvataan projektin johdon tavoitteet ja prioriteetit, ja oletukset riippuvuudet ja rajoitteet. Näissä kohdissa motiivitekijöihin kuuluvat ennen kaikkea projektin hallinnan filosofinen tausta ja tavoitteet, sekä lausunto ohjelmiston kehittämisestä, muokkaamisesta tai hankkimisesta. Projektin tavoitteisiin kuuluvat esim. budjetti, prioriteetit ja aikataulut. Prosessimäärittelyyn kuuluvat puolestaan raportoinnin mekanismit.

Kohdassa 3.3 kuvataan projektin riskien hallinta sisältäen riskitekijät, oletukset, riskien seurannan, ja suunnitelmat ongelmatilanteiden varalta.

Kohdat 3.4, 4.1, 4.2 ja 4.2 kuvaavat projektin prosesseja monitoroinnin, raportoinnin, ja teknisten prosessien kuten menetelmien, työkalujen, tekniikoiden ja dokumentoinnin osalta.

Kohta 5 kuvaa näkymän projektiin työkokonaisuuksien kannalta. Siinä kuvataan eri työkokonaisuudet, niiden riippuvuudet toisistaan ja ulkoisista tekijöistä, resurssien tarve ja jako, budjetti, ja aikataulut. Työkokonaisuudet käsitellään kokonaisuuden aliprojektien tapaan, ja niihin pätevät soveltuvin osin samat sitoumusmäärittelyt kuin koko projektiin.

Tarkasteltaessa IEEE 1058.1 standardia sitoumusten hallintamallin pohjalta se tarjoaa kohtalaisen hyvän alustan sitoumusten määrittelylle. Standardi on suhteellisen helposti kehitettävissä sisältämään tarvittavat sitoumukset. Nykyisessä muodossaan se ei kuitenkaan erityisesti ohjaa määrittelemään sitoumuksia, vaikka useimmat tärkeimmistä sitoumuksista sen avulla saadaan helposti kuvattua. Standardin mukainen projektisuunnitelma määrittelee parhaiten sitoumusten hallintamallin alueista projektin tavoitteet ja prosessit, mutta jättää riskien ja ongelmatilanteiden hallinnan ja varsinkin motiivitekijät vähemmälle huomiolle.

PRINCE2, IT-projektinhallinnan menetelmä

PRINCE on 1989 julkistettu IT projektien hallinnan menetelmä, joka on kehitetty 1970-luvulla muodostetun PROMPT- menetelmän pohjalta. Se kehitettiin Ison-Britannian valtion tietojenkäsittelyn ja tietoliikenteen hallinnon CCTA:n rahoittamana, ja se on tällä hetkellä Ison-Britannian valtion projektin hallinnan standardi. Sen nykyinen versio PRINCE2 julkistettiin 1996. PRINCE2 on prosessisuuntautunut, se keskittyy kysymään enemmänkin ”mitä” ja ”miksi” kuin ”miten”. Sitä sovelletaan monenlaisiin IT- ja muihin projekteihin ja se sisältää myös elementit asiakas/toimittaja suhteen hallintaan projektin aikana.[Bra97]

PRINCE2 jakaa projektin vaiheisiin, jotka ovat hallinnollisesti erillisiä. Kuten projekti, kukin vaihe on puolestaan jaettu prosesseihin, niillä on määritellyt tuotokset ja toiminnot, rajallinen kesto, omat kontrollinsa, ja oma organisaationsa. Haluttujen tuotosten valmistuminen ennalta määriteltujen laatuvaatimusten mukaan päättää kunkin hallinnollisen vaiheen.

PRINCE2 määrittelyt sisältävät:

- a) Projektin ja sen vaiheiden organisaation
- b) Projektia ohjaavat prosessit
- c) Projektisuunnitelman rakenteen ja sisällön
- d) Projektinhallinnan perustekniikan
- e) Joukon ohjauskeinoja, joilla projektin edistymistä voidaan valvoa

PRINCE2 määrittelee joukon hallinnollisten vaiheiden ajallisia kiinnekohtia, joiden tarkka ajankohta määritellään kullekin projektille sopivaksi. Nämä ajankohdat määräytyvät seuraavien tekijöiden mukaan:

- a) Tuotosten tuottamisen järjestys

- b) Tuotosten tai prosessien ryhmittely kokonaisuuksiksi
- c) Luontevat päätöksenteon pisteet katsauksille ja resurssien sitouttamiselle
- d) Projektin riskit ja liiketoiminnalliset vaikutukset
- e) Yhden tai useamman erillisen prosessin valmistuminen

Projektin vaiheet vastaavat projektin elinkaaren ominaisia askeleita kohti projektin tavoitetta. Tästä syystä ajalliset kiinnekohdat tavallisesti osuvat tärkeimpien tuotosten valmistumiseen, jolloin myös tehdään päätökset seuraavista vaiheista. Projektin luonteesta riippumatta on yleensä hyödyllistä perustaa yksi tai useampia prosesseja suunnittelua ja määrittelyä varten projektin alussa. Tätä varten PRINCE2 sisältää kaksi prosessia: ”Projektin perustaminen”, jossa luodaan perusta päätöksentelle, ja ”Projektin käynnistäminen”, jossa haetaan mm. ylemmän johdon sitoutuminen projektille.

PRINCE2:n rakenne on luonteva sitoumusten hallintaa varten. Erityisesti kunkin hallinnollisen vaiheen lopussa, mutta myös prosessien määrittelyissä on valmiit ajankohdat sitoumusten tarkistamiseen ja uusien tekemiseen. Vaikka PRINCE2 ei sinänsä erityisesti sisällä sitoumusten hallintaa muuten kuin projektin seuraavan vaiheen vaatimien resurssien osalta, se voidaan laajentaa koskemaan myös muita sitoumuksia. Pohja näille laajennuksille tulee projektin sopimuksesta, ja on siksi sovellettava kullekin projektille erikseen.

4 Empiirinen tutkimus

Tässä luvussa esitellään kirjallisuusselvityksen tueksi tehdyn empiirisen tutkimuksen tavoitteet, menetelmät, ja materiaalit. Tutkimuksen materiaali, eli tutkittavat projektit, esitellään tarkemmin luvussa 5.

4.1 Empiirisen tutkimuksen tavoitteet

Tässä työssä tehdyllä empiirisellä tutkimuksella pyrittiin löytämään vahvistusta kirjallisuusselvityksen ja kirjoittajan kokemuksen pohjalta tehtyyn tärkeimpien tekijöiden ja niiden vaikutuksen erittelyyn. Samalla pyrittiin löytämään vahvistusta tukemaan Kontion, Sulosen ja Pitkäsen mallia.

Lisäksi tavoitteena oli löytää eri tyyppisille projekteille olennaisimmat sitoumuksiin vaikuttavat tekijät vahvistukseksi projektien luokittelulle. Projektin alustava luokittelu perustettiin kirjoittajan omaan kokemukseen, ja empirialla haluttiin todeta, onko näillä projektiluokilla todennettavissa olevia ominaislaatuksia säännönmukaisuuksia tekijöiden vaikutuksesta. Näitä heuristiikkapatterneita tullaan käyttämään projektin sitoumusten ohjeiston luomiseen.

4.2 Tutkittavien projektien valinta

Sun Microsystems Oy:n asiantuntijapalvelut tekevät asiakkailleen ohjelmistoprojekteja varsin laajalla skaalalla. Projektit vaihtelevat puhtaasta arkkitehtuuri- ja teknologiakonsultoinnista valmisohjelmistojen muokkaamiseen ja soveltamiseen, ja kokonaan uusien sovellusten kehittämiseen. Tutkittavat projektit pyrittiin valitsemaan siten, että tämän skaalan tyypillisimmät projektit ovat mukana. Tämä valinta tehtiin myös siksi, että saataisiin projektien luokittelulle tarvittavaa pohjaa.

Projektien valinnassa kiinnitettiin myös huomiota siihen, että niistä oli saatavilla riittävästi kirjallista materiaalia sitoumusten selvittämiseksi. Vaikka kirjoittaja on ollut näissä hankkeissa osallisena varsinkin myyntivaiheessa, ja pystyi siksi täydentämään kirjallista materiaalia, annettiin pääpaino kirjallisessa materiaalissa kirjatuille seikoille. Kuten on aikaisemmin mainittu, henkilöiden muisti on valikoiva ja lyhyt, ja juuri siksi projekteissa on olennaista kirjata etenkin sitoumusten kaltaiset seikat tarkkaan.

4.3 Käytetyt tutkimusmenetelmät

Tutkimus suoritettiin post-mortem case study:na, joka perustui projektien sopimusten ja palaveripöytäkirjojen peilaamiseen mallin vaatimuksiin.

Tutkimusta varten kuvailtiin aluksi kunkin projektin hyvin ja huonosti menneet puolet subjektiivisena listana, jolle ei haettu yhteyttä sitoumusten hallintamalliin. Tähän listaan pyydettiin myös kommentteja projekteissa merkittävästi mukana olleilta henkilöiltä.

Tämän jälkeen kunkin projektin materiaalista kerättiin tehdyt sitoumukset ja niihin vaikuttaneet tekijät sitoumusten hallintamallin mukaan. Tätä verrattiin kirjallisuusselvityksen perusteella syntyneeseen listaan olennaisista sitoumuksista ja niihin vaikuttavista asioista. Näin pyrittiin

selittämään alussa tehtyä hyvin ja huonosti menneiden asioiden listaa. Samalla etsittiin myös vahvistusta kirjallisuusselvityksen perusteelle tehtyyn listaan, ja siten kehittämään listaa edelleen.

4.4 Empiirisen tutkimuksen ongelmat ja luotettavuus

Projektien tutkiminen post-mortem tapaustutkimuksena asettaa tiettyjä rajoitteita tutkimukselle. Joistakin projekteista oli tutkimusta tehdessä kulunut jo jonkin aikaa, ja mielikuvat sekä projektien vaiheista että ratkaisujen perusteista ovat todennäköisesti jossain määrin vääristyneet ja kaventuneet. Lisäksi projektin tulkitseminen pelkkien projektisopimusten ja pöytäkirjojen perusteella antaa potentiaalisesti melko kapean näkymän projektin vaiheisiin ja taustoihin.

Kirjoittajan itsensä osallistuminen projektien joihinkin vaiheisiin saattaa myös asettaa tulkintojen objektiivisuuden kyseenalaiseksi, varsinkin silloin kun joudutaan selvittämään puutteita toimittajan tekemisissä. Tämän tekijän vaikutusta pyrittiin minimoimaan peilaamalla projektin tapahtumat yksikäsitteisesti sitoumusten hallintamallin aihealueisiin ja sitoumuksiin vaikuttavien tekijöiden kuvauksiin. Tutkimuksessa pyrittiin välttämään potentiaalisesti subjektiivisten arvioiden tekoa silloin kun se oli mahdollista. Toisaalta kirjoittajan kosketuspinta projekteihin auttaa taustojen selvittämisessä, ja siten johtaa nopeammin oikeille jäljille.

Objektiivisuuden lisäämiseksi projekteista pyydettiin lyhyitä arvioita niissä kiinteämmin mukana olleilta henkilöiltä. Varsinaisia haastatteluja ei ollut aikarajoitteiden takia mahdollista tehdä, mutta nämä lyhyetkin arviot opastivat osaltaan oikeaan suuntaan projektien arvioinnissa.

Tutkittava materiaali ei todennäköisesti ollut myöskään niin laaja, kuin tällaisessa tutkimuksessa olisi suositeltavaa. Projekteja tutkittiin kuusi kappaletta, ja jokaiseen projektiluokkaan saatiin vähintään yhden projektin materiaali. Tosin näinkin suppea materiaali riitti tavoitteena olleen kvalitatiivisen tutkimuksen tekemiseen.

4.5 Projektien luokittelu

Sun Microsystems Oy:n asiantuntijapalvelujen ohjelmistoprojektit ovat tyypillisesti joko konsultointia, valmisohjelmistojen muokkaamista, tai kokonaan uuden sovelluksen kehittämistä. Tässä työssä projektien luokittelu perustettiin näille kolmelle perustyyppille. Kokemuksen mukaan ne poikkeavat toisistaan monessa suhteessa: vastuuasiat määritellään eri tavoin, osapuolten odotukset ovat erilaiset, asiakas/toimittaja- suhteen laatu ja syvällisyys ovat eri tasoilla jne.

Monet projektit sisältävät elementtejä useammasta kuin yhdestä luokasta. Jos eri luokkiin sijoittuvat projektin osat ovat eroteltavissa esimerkiksi ajallisesti, ne voidaan käsitellä erikseen. Muussa tapauksessa on päätettävä, mihin luokkaan projekti olennaisilta osiltaan kuuluu ja käsitellä sitoumukset sen mukaan, ottaen huomioon muiden luokkien asettamat vaatimukset.

Puhtaisiin konsultointihankkeisiin toimittajan konsultit osallistuvat teknisinä neuvonantajina, joiden tehtävä on ohjata varsinkin projektin tekninen toteutus oikeaan suuntaan. Konsulttien arvo asiakkaalle tulee siitä, miten

hyvin he osaavat päätellä oikean suunnan projektin tavoitteista, ja miten hyvin he osaavat ohjata projektin siihen suuntaan eri tilanteissa.

Konsultointihankkeissa toimittajalla ei ole suoraa sopimuksellista vastuuta tuotetun ohjelmiston laadusta ja projektin yleisestä menestyksestä. Suurimmat riskit näissä hankkeissa ovat seurausta siitä, että toimittajalla on suuri implisiittinen vastuu suoran sopimusvastuun puuttumisesta huolimatta. Kun lisäksi toimittajan konsulteilla ei ole suoraa vaikutusmahdollisuutta projektin toteuttajiin, on toimittajan asema monessa suhteessa riskialttiimpi kuin muunlaisissa hankkeissa, varsinkin kun toimittaja ei ongelmatilanteessa ole oikeutettu korvauksiin yhtä kattavasti kuin niissä.

Projekteissa, joiden pääsisältö on valmisohjelmistojen soveltaminen, joko rakennetaan ohjelmisto valmiskomponenttien pohjalta tai muokataan sellaisenaan toimiva, mutta muokattavaksi suunniteltu ohjelmisto asiakkaan tarpeiden mukaan. Sekä valmiskomponentit että muokattava ohjelmisto ovat kehitetty tiettyä soveltamisaluetta varten, kuten portaali-ohjelmistoiksi tai sähköisen kaupankäynnin perusohjelmistoiksi. Koska suurin osa koodista saadaan valmiina, on projektin tavoitteena oleva ohjelmisto suhteellisen hyvin määritetty. Toimittaja pystyy myös tekemään muita hankkeita luotettavamman arvion työmääristä, kustannuksista, ja ennen kaikkea projektin riskeistä, koska projektin tehtäväalue on hyvin tunnettu.

Näissä projekteissa toimittajan riskit riippuvat siitä, miten suuria laajennuksia ja muutoksia ohjelmistoon tarvitaan. Jos muutokset ovat suuria, projekti lähestyy kokonaan uuden ohjelmiston kehittämistä, ja olisi käsiteltävä sellaisena myös sitoumusten kannalta.

Uuden sovelluksen kehittämisprojekteilla tarkoitetaan perinteistä ohjelmistoprojektia, jossa asiakkaan tarpeiden pohjalta luodaan ohjelmisto käytännössä alusta asti. Joitain perusteknologioita käytetään, mutta ne ovat huomattavasti geneerisempiä kuin varsinaisia valmisohjelmistoja sovellettaessa. Tällaisia voivat olla esimerkiksi Java 2 Enterprise Edition perusolioineen, tai käyttöliittymän luomiseen tarvittavat oliokirjastot.

Kokonaan uutta ohjelmistoa kehitettäessä tehdään laajimmat sitoumukset, olivat ne sitten kirjattu sopimukseen tai ei. Koska valmista sovellusalueelle tarkoitettua koodia ei käytetä, ei projektin vaatimaa panostusta ja riskejä tunneta tarkkaan.

5 Tutkimuksen analyysi

Tässä luvussa käsitellään sitoumuksiin vaikuttavat tekijät aikaisemmin esitetyn kirjallisuuden, tutkimuksen ja kirjoittajan oman kokemuksen perusteella. Kunkin tekijän vaikutus peilataan sitoumusten hallintamallin neljään aihealueeseen: motiivitekijöihin, projektin tavoitteisiin, prosessien määrittelyyn, ja riskien ja ongelmatilanteiden hallintaan.

Tekijöistä on valittu kahdeksan tutkimuksen perusteella tärkeimmäksi havaittua. Valintaan vaikutti sekä se miten usein niihin viitattiin, niiden painoarvo eri lähteiden ja tutkimuksen mukaan, ja tekijän oma näkemys.

Näiden kahdeksan lisäksi löydettiin todisteita muidenkin tekijöiden vaikutuksesta tietyissä tilanteissa, mutta ne jätettiin tässä työssä käsittelemättä tarkemmin. Näitä olivat esimerkiksi tietoturva vaatimukset, käytetyt toimitusehdot, ohjelmiston elinkaaren mallit ja rinnakkaisuus, johdon sitoutuminen, projektin organisointi, mittausten menetelmät ja tavoitteet, tekijänoikeudelliset seikat, sekä projektin jälkeinen tuki ja takuu.

5.1 Tekijöiden vaikutukset sitoumusten hallintamallin mukaisesti

5.1.1 Projektin koko

Projektin koko voidaan mitata kuhunkin tapaukseen sopivalla yhdellä tai useammalla mittarilla, kuten projektiin osallistuvien henkilöiden lukumäärä, projektin suunnitellut kustannukset, tai tuotetun ohjelmiston koko ohjelmariveinä tai toimintoina. Tärkeintä on valita ne mittarit, joilla on ensisijaisesti merkitystä organisaation kannalta, ja vasta toissijaisesti projektin kannalta, koska projektin henkilömäärä tai arvioitu laskutus ei kerro koko totuutta projektin merkityksestä.

Motiivitekijät

Pääsääntöisesti suurella projektilla on enemmän liiketaloudellista merkitystä kuin pienellä. Suurella projektilla pyritään joko liiketoiminnan tehostamiseen tai kilpailuedun tavoittamiseen, joten suuren projektin strateginen painoarvo on tyypillisesti suurempi kuin pienen. Joissakin tapauksissa pienelläkin projektilla voi olla suuri painoarvo, jolloin niidenkin motiivien määrittely on olennaista.

Kaikissa projekteissa motiivitekijöiden määrittelemine on tärkeää, mutta tärkeys korostuu projektin koon kasvaessa. Motiivien määrittelemine on olennaista etenkin projektiin osallistuvien henkilöiden kannalta, koska vaikka projektin johto varmasti ymmärtääkin ne, jokaisen projektissa työskentelevän on myös tärkeää ymmärtää ne.

Projektin tavoitteet

Suuret projektit jaotellaan tavallisesti pienempiin osaprojekteihin, joille määritellään omia osatavoitteita. Nämä tavoitteet voivat olla joko osakokonaisuuksien valmistumisia, ajallisia, tai yhdistelmiä näistä.

Prosessien määrittelyt

Prosessien määrittely tulee tärkeämmäksi, kun projektin koko kasvaa. Varsinkin projektin hallinnan lisätyö kasvaa, koska projektin kokonaiskuvaa on vaikeampi hahmottaa.

Riskien ja ongelmien hallinta

Projektin koon kasvaminen tuo lisähaasteita riskien ja ongelmatilanteiden hallinnalle. Suurissa projekteissa riskit ja sen kautta ongelmatilanteet on pyrittävä eristämään siten, että niiden vaikutus rajautuu mahdollisimman pieneen osaan kokonaisprojektia.

Riskien sieto puolestaan kasvaa projektin koon kasvaessa. Pienellä projektilla ei ole suurta liikkumavaraa aikataulun venymiselle tai kustannusten kasvulle, kun taas suurempi projekti voi kestää ne paremmin.

5.1.2 Osapuolten lukumäärä ja organisoituminen

Tarvittaviin sitoumuksiin vaikuttaa sekä projektiin osallistuvien osapuolten lukumäärä että se tapa, jolla ne organisoidaan. Asiakas voi jakaa projektin eri osat eri osapuolille, ilman että niillä on keskinäisiä vastuuta ja tavoitteita, tai vastuut voidaan antaa yhdelle integroivalle osapuolelle, joka edelleen jakaa projektista osia tehtäviksi alihankkijoilleen. Jokaisessa hankkeessa on kuitenkin aina yksi osapuoli, jolla on kokonaisvastuu projektista. Jos vastuu on jaettu usealle toimittajalle, kokonaisvastuu jää asiakkaalle itselleen. Lisäksi organisoinnilla asiakas voi valita mitä vastuuta se siirtää edelleen ja mitä se jättää itselleen, sillä vastuun mukana siirtyy myös merkittävä osa projektin kontrollia.

Mitä enemmän osapuolia projektiin osallistuu, sitä suurempi merkitys sitoumuksilla on. Sitoumuksilla hallitaan projektiin osallistuvien osapuolten verkostoa, sidotaan toimittajat toisiinsa ja asiakkaaseen, ja myös asiakas toimittajiin.

Kahden osapuolen (kuten asiakas/toimittaja) projektissa sitoumukset ovat myös huomattavasti helpommin tunnistettavissa kuin useamman osapuolen hankkeissa.

Motiivitekijät

Kun projektin osapuolten lukumäärä on pieni, eri osapuolten tavoitteet ja motiivitekijät ovat suhteellisen ilmeisiä. Kun lukumäärä kasvaa, tavoitteet jäävät helposti epämääräisemmiksi. Esimerkiksi kun yksi tai useampia alihankkijoita osallistuu yhden integroivan osapuolen kanssa asiakkaalle toimitettavaan projektiin, saattaa alihankkijoiden taustalla olevat motiivit olla hyvin vaihtelevat. He saattavat hakeutua projektiin kehittääkseen suhdetta integraattoriin, asiakkaaseen tai molempiin.

Erityisesti jos useat osapuolet toimivat rinnakkain samassa projektissa, eri osapuolten motiivitekijät jäävät usein varsinkin toisilleen suurelta osin arvailujen varaan. Tällaisessa tapauksessa motiivitekijöiden määrittelemisen onkin asiakkaan vastuulla.

Projektin tavoitteet

Usean osapuolen projekteissa osa projektin tavoitteista on jaettava osapuolten kesken, osa taas on kaikille yhteisiä. Teknisiin tavoitteisiin, aikatauluihin ja kustannuksiin liittyvät tavoitteet on tavallisesti mahdollista jakaa, mutta yleiset tavoitteet ja rajoitteet jäävät jokaiselle osapuolelle yhteisiksi.

Jos tavoitteet jaetaan osiin, jonkin osapuolen on otettava vastuu niiden kaikkien kokoamisesta. Tämä osapuoli voi olla joka asiakas itse tai integroivana osapuolena toimiva toimittaja.

Kokonaistavoitteisiin liittyvät sitoumukset on määriteltävä selkeästi silloinkin, kun asiakas itse toimii integroivana osapuolena. Näin sekä asiakkaan omat projektiin osallistuvat henkilöt että toimittavat osapuolet ymmärtävät ne selkeämmin.

Prosessien määrittelyt

Prosessien määrittelysten merkitys korostuu projektin osapuolten lukumäärän kasvaessa. Tämä koskee ennen kaikkea osapuolten keskinäisiä prosesseja, joilla koko projekti kootaan yhteen, mutta myös osapuolten sisäisiä prosesseja. Dokumentoimalla osapuolten sisäiset prosessit soveltuvin osin parannetaan niiden läpinäkyvyyttä projektin osapuolten kesken, ja yhteiset prosessit saadaan toimimaan luotettavasti.

Mitä suurempi määrä osapuolia projektiin osallistuu, sitä olennaisempaa yhteisten prosessien määrittely on ja sitä selkeämmin osapuolten on sitouduttava tiettyihin ydinprosesseihin, kuten kehityksen prosessimalliin, projektin hallintaan kokouksineen, riskien hallinnan prosesseihin ja laadunvalvontaan.

Riskien ja ongelmien hallinta

Riskien hallintaan pätee jälleen sama periaate kuin vastuun jakoon: jokaisessa projektissa yhdellä osapuolella on kokonaisriskit kannettavanaan. Tosin toisin kuin vastuuta jaettaessa, yksikään osapuoli ei voi täysin välttyä riskeiltä. Varsinkin asiakkaalla on aina tietty riski projektiin liittyen, ja se on sitä suurempi mitä enemmän asiakkaan liiketoiminta riippuu projektin onnistumisesta.

Projektin riskejä on kuitenkin mahdollista jakaa, ja näin yleensä onkin suositeltavaa tehdä. Jos jollain osapuolella ei ole mitään merkittävää riskiä kannettavanaan, hänen panostuksensa ja sitoutumisensa projektiin voidaan asettaa kyseenalaiseksi. Riskien jakaminen eri osapuolille vaikuttaa hyvin merkittävästi motivaatioon sitoutua projektiin.

Projektiin osallistuvien osapuolten lukumäärä ei merkittävästi vaikuta riskien hallinnan tärkeyteen, koska se on joka tapauksessa yksi olennaisimmista määriteltävistä elementeistä. Osapuolten määrän kasvaessa riskien hallinta kuitenkin monimutkaistuu.

Ongelmatilanteiden hallinta on myös tärkeää riippumatta projektin osapuolten lukumäärästä, mutta tilanteiden kommunikoinnin merkitys kasvaa osapuolten määrän kasvaessa. Lisäksi organisointitapa vaikuttaa kommunikoinnista sopimisen tärkeyteen. On mahdollista, että jokin

alihankkijoista törmää omassa osuudessaan ongelmiin, ja tiedottaa niistä integroivalle osapuolelle, mutta tieto ei saavuta asiakasta tai muita projektin osapuolia.

5.1.3 Ulkopuolinen vai sisäinen toimittaja?

Vaikka tässä tutkimuksessa keskitytään erityisesti sellaisiin projekteihin, joissa ohjelmisto tilataan ulkopuoliselta toimittajalta, on tutkimuksen kannalta kiinnostavaa myös selvittää millaisia eroja se aiheuttaa projektin sitoumuksiin. Oletus on, että sitoumukset ovat pääosin samat molemmissa tapauksissa, mutta jotkin olennaiset niihin vaikuttavat tekijät, kuten liiketoimintamalli puuttuvat kun projekti toteutetaan asiakkaan omin resurssein.

Yritysten kokonaisliiketoiminnan seurannan ansiosta sisäisetkin projektit useimmiten tilataan sisäiseltä organisaation osalta samalla tavalla kuin jos se tilattaisiin ulkoiselta toimittajalta. Vaikka näin voidaan projektia seurata tarkemmin, tilanne on merkittävästi erilainen kuin tilattaessa ulkopuoliselta. Oman organisaation toimittajalla on kiinteät suhteet asiakkaaseen, korkealla tasolla tavoitteet ovat yhteiset, ja todellista laskutusta ei suoriteta.

Suurin ero sisäisellä ja ulkoisella toimittajalla on juuri se tosiasia, että ulkoinen toimittaja on huomattavasti riippuvaisempi saamastaan laskutuksesta kuin sisäinen. Sisäisen toimittajan saama laskutus on selkeimmilläänkin vain kirjanpidollinen toimenpide yrityksen sisällä. Tästä syystä sisäinen ja ulkoinen toimittaja myös sitoutetaan hieman eri tavalla, ja erilaisiin sitomusalueisiin.

Sisäinen toimittaja sitoutetaan parhaiten motiivitekijöiden kautta, ulkoinen projektin tavoitteiden kautta. Molemmat saadaan sitoutumaan prosesseihin ja riskien ja ongelmien hallintaan.

Motiivitekijät

Koska sisäinen toimittaja on samaa yritystä kuin asiakas, heillä on todennäköisesti korkealla tasolla joitakin yhteisiä tavoitteita. Näistä puolestaan saadaan johdettua hankkeelle motiivitekijät, joilla on molemmille osapuolille merkitystä.

Ulkoista toimittajaa on vaikeampi sitouttaa motiivitekijöihin. Jos yhteistyön ei ole tarkoitus kestää kuin projektin ajan, ei todellisia yhteisiä strategisia tarkoituksia tai missioita ole kovin helppo tunnistaa. Jos taas suhteen on tarkoitus olla pitkäikäinen ja kehittyä syvälliseksi, on motiivitekijöiden määrittäminen hyvin olennaista, jotta molemmat tunnistavat mitä suhteelta halutaan pitkällä aikavälillä.

Projektin tavoitteet

Tilattaessa ohjelmisto ulkopuoliselta toimittajalta voidaan monet projektin tavoitteista sitoa esimerkiksi laskutukseen. Tällä tavalla ulkoisen toimittajan tekemät tavoitteisiin liittyvät sitoumukset muodostuvat varsin vahvoiksi.

Vastaavaa yhteyttä ei voida rakentaa, jos toimittava osapuoli on sisäinen, koska laskutus ei ole silloin todellista. Sisäisen toimittajan sitoutuminen projektin tavoitteisiin on saavutettava muilla keinoilla.

Prosessien määrittelyt

Osapuolet sitoutuvat molemmissa tapauksissa samalla tavalla määriteltäviin prosesseihin. Projektin kokouskäytännöt, kehitysprosessin määrittelyt, raportoinnit, muutosten hallinta ja riskien hallinnan prosessien määrittelyt eivät olennaisesti riipu siitä, onko toimittaja ulkopuolinen vai samaa yritystä. Tärkein poikkeus on tilaajan ja ulkoisen toimittajan välisten prosessien määrittäminen laskutukseen liittyviltä osiltaan.

Riskien ja ongelmien hallinta

Asiakas ja toimittaja pyrkivät tavallisesti jakamaan projektiin liittyvät riskit, siten että molemmille jää kannettavaksi sellaiset riskit, joiden välttämiseen he pystyvät parhaiten vaikuttamaan. Lisäksi riskit pyritään arvottamaan, ja määrittelemään sopimukseen korvauksia niiden laukeamisesta.

Sekä riskien jakaminen että varsinkin niihin liittyvien korvauksien määrittäminen on selkeämpää silloin, kun toimittaja on ulkopuolinen. Molemmat voidaan tietysti määritellä myös sisäisen toimittajan tapauksessa, mutta taas tässäkin tapauksessa todellisen laskutuksen puuttuminen tekee sitoumuksista heikompia.

5.1.4 Projektin aikataulu

Sitoutuminen aikatauluun tarkoittaa ennen kaikkea osapuolten sitoutumista siihen, että tietyt tuotokset ovat valmiita määritellyn aikataulun mukaisesti. Näitä tuotoksia voivat olla esimerkiksi projektin alussa tarvittavien materiaalien, välineiden ja tilojen käyttöön saaminen, projektin aikana osakokonaisuuksien valmistuminen ja uusien tarvittavien resurssien hankinta, ja projektin lopussa kohteena olleiden ohjelmistojen ja dokumenttien toimittaminen.

Aikataulu vaikuttaa sitoumuksiin kahta kautta: projektin vaiheiden tahdistuksen ja ajastuksen kautta. Tahdistus tarkoittaa eri työvaiheisiin kuluva aikaa, ja ajastus puolestaan sitä ajankohtaa, jolloin vaihe aloitetaan tai päätetään.

Motiivitekijät

Projektin kriittisyyden tärkeimpiä indikaattoreita on se, että sen valmistuminen suunnitelmien mukaan ajallaan on kriittistä asiakkaan muulle liiketoiminnalle. Tällaisessa tilanteessa sekä asiakkaan että toimittajan sitoutuminen aikatauluun on tehtävä riittävän voimakkaaksi. Pohja sitoumuksille on tällaisten kriittisten aikataulullisten motiivitekijöiden tunnistaminen ja kirjaaminen sopimuksiin.

Jos hanke ei ole liiketoiminnan aikatauluille kriittinen, ei sitoumuksia tarvitse tehdä yhtä kiinteästi. Eri osapuolten sitoutuminen projektin aikatauluun on kuitenkin joka tapauksessa eräs keskeisimmistä sitoumusalueista.

Projektin tavoitteet

Aikataulu on jo sinällään yksi tärkeimmistä projektin tavoitteista. Sillä on usein myös vaikutuksia muihin tavoitteisiin, kuten kuluihin ja rajoitteisiin. On mahdollista, että projektin kireän aikataulun takia siihen joudutaan ottamaan ylimääräisiä resursseja tai päädytään käyttämään sellaisia tuotteita ja teknologioita, jotka nopeuttavat kehitystä mutta nostavat kustannuksia.

Lisäksi aikataulun tahdistus vaikuttaa projektin tuotoksiin. Nopeassa aikataulussa esimerkiksi ei usein ole mahdollista tuottaa kaikkea tarvittavaa toiminnallisuutta. Tällaisissa tapauksissa projekti yleensä jaetaan vaiheisiin, jotta kriittisimmät osat ohjelmistosta saadaan mahdollisimman pian käyttöön.

Lisäksi projektin ajastus vaikuttaa tavoitteisiin. Jotkin tarvittavat resurssit tai teknologiat saattavat tulla käyttöön vasta jonain tiettyinä ajanhetkenä, tai asiakas ei voi ottaa valmistuvia ohjelmiston osia testikäyttöön ennen jotain ajankohtaa.

Prosessien määrittelyt

Projektin edistymisen seuranta on eräs keskeisimmistä projektin hallinnan tavoitteista. Hankkeen tila aikajanalla on kyettävä tunnistamaan, jotta tarvittaviin toimenpiteisiin voidaan ryhtyä, jos aikataulusta ollaan jäljessä.

Osapuolten on sovittava keskenään menettelyt aikataulun muuttamiseen. Tarve muutoksille voi tulla miltä tahansa osapuolelta, joten menettely on sovittava yhteiseksi, samoin muutoksista on sovittava yhteisesti. Samalla on tunnistettava, miten muutokset vaikuttavat koko hankkeen aikatauluun, ja muutettava sitoumukset vastaamaan uutta aikataulua.

Riskien ja ongelmien hallinta

Projektin aikataululla on merkittävä vaikutus myös sen riskeihin. Liian kireäksi neuvoteltu aikataulu sekä vaarantaa ohjelmiston laadun että sisältää itsessään riskin aikataulun lipsumisesta. Lisäksi on mahdollista, että jokin tuote tai teknologia ei ole käytettävissä sinä ajanhetkenä kuin oli suunniteltu.

Lyhyessä projektissa riskit ovat usein merkittävästi pitkiä projekteja suurempia, koska aikataulussa ei ole joustovaraa. Pitkissä projekteissa on puolestaan mahdollista ottaa jossain kohdassa menetetty aika kiinni jossain toisessa vaiheessa. Aikatauluihin liittyvät riskit esimerkiksi projektin kriittisen polun avulla on syytä tunnistaa jo projektin sopimuksessa ja projektisuunnitelmissa.

Aikataulusta lipsumisen aiheuttamat toimenpiteet on myös syytä määritellä jo projektin sopimuksessa, samoin kuin aikatauluihin liittyvät oletukset.

5.1.5 Liiketoimintamalli

Varsinkin kun ohjelmistoprojekti tilataan ulkopuoliselta toimittajalta, sekä asiakkaan että toimittajan kannalta tärkein sitouttamisväline on projektin liiketoimintamalli. Koska viime kädessä yksi tärkeimmistä projektin onnistumisen mittareista on rahallisten tavoitteiden täyttyminen, on

erilaisilla rahallisilla kepeillä ja porkkanoilla suuri vaikutus koko projektin onnistumiseen. Asiakas haluaa varmistaa, että hän saa rahoilleen joka tilanteessa vastinetta. Toimittaja puolestaan haluaa maksimoida tuottonsa hankkeesta ja varmistaa, että hankkeen kassavirrat ovat suunnitellun mukaiset.

Ohjelmistoprojektin liiketoimintamalleja on useita erilaisia, ja joukkoa rajoittaa lähinnä asiakkaiden ja toimittajien mielikuvitus ja halukkuus riskinottoon. Yksinkertaisin malli on kiinteähintainen projekti, jossa asiakas suorittaa ennakoon sovitun kiinteän summan, kun sovittu kokonaisuus on valmis. Tässä mallissa riskit painottuvat toimittajalle, koska hankkeen kannattavuus riippuu täysin toimittajan kyvystä sekä arvioida hankkeeseen tarvittavat resurssit ja niiden kulut, että toimittaa projekti odottamattomista hankaluuksista huolimatta suunnitellulla tavalla. Usein varsinkin suuremmissa hankkeissa osa riskistä siirretään asiakkaalle kiinnittämällä laskutuseriä projektin osakokonaisuuksien valmistumiseen. Tässä mallissa toimittaja voi laskuttaa asiakasta tehdystä työstä, vaikka projektin kokonaistavoitteet eivät täyttyisi.

Toimittajalla on pienin riski ja sitoutuminen hankkeissa, joiden laskutus perustuu kokonaan tunti- tai päivähintoihin. Yksinkertaisimmassa mallissa toimittaja myy asiakkaalle tekijöitä, joiden osaamistaso on taattu, mutta toimittaja sitoutuu vain heikosti projektin lopputuloksen tuottamiseen. Useimmiten tähän malliin yhdistetään arviot työhön kuluvasta ajasta, minimi- ja maksimilaskutukset ja laskutuksen jaksottaminen.

Kokonaan toisenlainen malli on laskutus sitominen hyötyyn, jonka asiakas saa ohjelmistosta. Toimittajan laskutus voi olla esimerkiksi osuus saavutetuista säästöistä tai lisätuotoista profit sharing- periaatteella. Lisäksi jos tuotettu ohjelmisto on osa tuotetta, jota asiakas myy edelleen, osa laskutuksesta saatetaan jaksottaa näihin toimitettuihin tuotteisiin.

Rahallisen laskutuksen lisäksi toimittaja saattaa saada muita hyötyjä tuotetusta ohjelmistosta. Strategisesti tärkeissä projekteissa saatetaan esimerkiksi arvottaa toimittajan projektista saama markkinoinnillinen hyöty. Toimittaja saattaa myös ottaa lisäriskejä tai jopa subventoida ohjelmistoprojektia saadakseen ohjelmistoprojektin avulla jonkin muun ohjelmiston tai laitteiston myytyä asiakkaalle.

Usein projektin liiketoiminnallinen malli on yhdistelmä edellämainittuja. Malli saattaa jopa muuttua projektin edetessä, jolloin myös asiakkaan ja toimittajan sitoumukset vaihtelevat. Kaikissa tapauksissa liiketoiminnallinen malli vaikuttaa merkittävästi siihen, millaisia asioihin asiakas ja toimittaja sitoutuvat projektissa.

Motiivitekijät

Projektin motiivitekijät vaikuttavat merkittävästi sen liiketoimintamalliin. Tyypillisesti suoraviivaisissa, strategisesti vähemmän tärkeissä hankkeissa, joissa esimerkiksi muokataan olemassa olevaa ohjelmistoa asiakkaan tarpeisiin, sovitaan tunti- tai päivähinnoittelusta, kun taas strategisesti tärkeissä ja haastavissa hankkeissa asiakas haluaa siirtää mahdollisimman suuren osa riskeistä toimittajalle.

Vähemmän strategisissa projekteissa toimittajan sitoumukset ovat kevyimmät. Toimittaja sitoutuu lähinnä takaamaan resurssien määrän, laadun, ja aikataulun. Usein toimittaja kuitenkin sitoutuu tämän lisäksi tiettyyn lopputulokseen. Asiakkaan sitoumukset ovat myös kevyet. Strategisissa hankkeissa toimittaja kantaa tavallisesti osan riskistä asiakkaan kanssa jossain muodossa.

Projektin tavoitteet

Projektin tilaaja pyrkii rakentamaan liiketoiminnallisen mallin siten, että toimittaja on mahdollisimman hyvin sitoutunut projektin tavoitteisiin. Laskutuserät sidotaan tiettyjen kokonaisuuksien valmistumiseen suunnitelman mukaan, ja toimittajalle on joko määrätty sanktioita tai ylimääräistä aikaa ei korvata. Toimittaja saadaan näin sitoutumaan tiettyihin tavoitteisiin varsin vahvasti.

Toisaalta myös toimittaja voi suojata itseänsä liiketoiminnallisen mallin avulla, ja saada asiakas sitoutumaan niihin tavoitteisiin, joiden pohjalta ohjelmistosopimus on tehty. Lisäksi toimittajan kannattaa määritellä sopimukseen palkkiot tavoitteiden ylittämisestä. Esimerkiksi jos ohjelmisto valmistuu tavoiteaikataulua nopeammin, toimittaja voi aikalaskutukseen perustuvassa projektissa veloittaa osan alituksesta.

Prosessien määrittelyt

Liiketoiminnallinen malli vaikuttaa merkittävästi eri osapuolten motivaatioon osallistua projektin prosesseihin ja niiden määrittelyyn. Sekä toimittajalla että asiakkaalla on selkeä insentiivi valvoa, että esim. projektin eri vaiheisiin sidotut palkkiot laskutetaan heti kun vaihe on yhdessä todettu toimitetuksi.

Prosesseja määriteltäessä toimittaja sitoutuu projektisopimuksen mukaisiin ehtoihin ja vaiheisiin laskutuksessa. Toimittajan onkin sopimusta laadittaessa harkittava tarkoin, miten nämä ehdot ja ennen kaikkea laskutuksen ajoitus sopii hänen omalle liiketoiminnalleen, koska varsinkin merkittävän kokoisen projektin laskutuksen ajoitus voi vaikuttaa paljon toimittajan kassavirtaan.

Asiakkaan on puolestaan sitouduttava niihin ehtoihin, joiden mukaan osalaskutusta saadaan tehdä. Usein ohjelmistojen tilaajille tulee projektin aikana kiusaus lisätä vaatimuksia jollekin projektin osalle ja asettaa samalla lisäehtoja laskutukselle. Asiakkaan on myös hyvä tunnistaa laskutuksen vaikutus toimittajan kassavirtaan, ja välttää ajamasta toimittajaa yllättävillä vaatimuksilla hankalaan tilanteeseen taloudellisesti.

Riskien ja ongelmien hallinta

Liiketoiminnallinen malli useimmiten sisältää myös provisiot erilaisille korvauksille ongelmatilanteissa. Kun riskit on tunnistettu projektista sovittaessa, niille voidaan määritellä myös rahallinen tai muu arvo, joka linjataan projektin liiketoiminnallisen mallin kanssa. Näiden lisäksi tavallisesti sovitaan myös ennakoimattomien ongelmatilanteiden korvauksesta. Sen varalta että projekti joutuu niin vakavaan ongelmatilanteeseen, että sen tärkeimpien tavoitteiden täytyminen

vaarantuu, sovitaan lisäksi mm. sopimuksen purkamisen yhteydessä tehtävistä korvauksista.

Koska taloudelliset keinot ovat voimakkaimmat ohjausvälineet myös ohjelmistoprojekteissa, projektin taloudelliset riskit on molempien kannalta edullista jakaa järkevällä tavalla. Jos asiakas siirtää kaiken riskin toimittajalle, toimittaja joutuu tasapainottamaan suuren riskin korkeammalla hinnalla, ja asiakas joutuu maksamaan enemmän silloinkin kun projektin riskit eivät toteudu. Vastaavasti jos toimittaja ei ota lainkaan riskiä, hän sitoutuu projektin tavoitteisiin suhteellisen heikosti, ja asiakas joutuu jälleen maksamaan lisää optimaaliseen tilanteeseen verrattuna.

Liiketoimintamalli vaikuttaa merkittävästi juuri riskeihin ja ongelmatilanteisiin liittyviin sitoumuksiin, koska ne ovat luonteeltaan taloudellisia. Nämä sitoumukset olisikin laadittava siten, että niillä pyritään vain korvaamaan mahdollinen menetys äärimmäisessäkin tilanteessa, eikä rankaisemaan mitään osapuolta.

5.1.6 Valmiskomponenttien käyttö

Ohjelmistotuotannon tehostamisen suurimpana lupauksena on pidetty ohjelmiston osien uudelleen käyttöä. Ohjelmistot rakennetaan yhä useammin valmiista komponenteista, ja itse tuotettua koodia halutaan käyttää mahdollisimman tehokkaasti jatkoprojekteissa. Vaikka valmiskomponenttien käytöllä saavutetaan monia etuja, tärkein syy niiden käytölle on ajan ja resurssien käytössä saatavat säästöt, koska ohjelmistojen tuottaminen alusta asti on hyvin kallista ja aikaa vievää. [Voa98a]

Valmiskomponenttien käyttöön liittyy kuitenkin monia riskejä. Yhden komponentin huono laatu saattaa riskeerata koko projektin, komponentin tietoturva saattaa olla heikompi kuin vaaditaan, tai voi olla että komponentti ei ylipäätään toimi oikein osana ohjelmistoa. [Voa98b]

Muista ohjelmistoprojektin toiminnoista poiketen, valmiskomponenttien hankinta ja käyttöönotto muistuttaa pitkälle tuotteen hankintaa. Projektista riippuen asiakkaalla voi kuitenkin olla enemmän tai vähemmän mahdollisuuksia saada toivomiaan muutoksia komponentteihin, ja saada toimittaja aktiivisesti mukaan ohjelmistoprojektiin. Näin on varsinkin sellaisissa projekteissa, joissa valmisohjelmisto muodostaa merkittävän osan projektin lopputuloksesta. Toimittajan saaminen mukaan ohjelmistoprojektiin on joka tapauksessa asiakkaalle edullista.

Motiivitekijät

Valmiin komponentin tuottajan tärkein motivaatio liittyy siihen tosiseikkaan, että ohjelmistojen tuottaminen on joka tapauksessa kallista. Tuotteistamalla komponentin tuottaja pyrkii saamaan kerran tehdystä työstä kompensaation useaan kertaan ja siten vähintään peittämään omat kustannuksensa ja mielellään tuottamaan myös voittoa. Tuotteistamiseen liittyy lisäinvestointeja, koska tuottajan on testattava ja dokumentoitava tuote perusteellisemmin kuin sisäisessä käytössä ja järjestettävä myynnin jälkeinen tuki jossain muodossa, mutta jos tuotetta onnistutaan myymään, nämäkin investoinnit saadaan katettua.

Valmiskomponentin tilaaja puolestaan odottaa saavuttavansa selkeää etua joko pienempien kustannusten tai nopeamman aikataulun muodossa. Hänen on toisaalta kyettävä varmistumaan siitä, että suunnitellut edut saavutetaan, toisaalta saatava provisiot sen varalle, että niitä ei saavuteta.

Tuottajan ja asiakkaan on määriteltävä, minkä takia valmiskomponentteja otetaan käyttöön, mitä etuja niiden käytöllä tavoitellaan. Samalla voidaan sopia lisäehdot, joilla sekä tuottaja että asiakas suojaavat itseänsä tämän määritelmän ulkopuolelle menevän käytön aiheuttamien ongelmatilanteiden varalta.

Projektin tavoitteet

Kuten aikaisemmin on mainittu, valmiiden komponenttien käytöllä pyritään joko säästämään kustannuksissa tai nopeuttamaan aikataulua. Nämä tavoitteet on kirjattava mahdollisimman tarkkaan sopimukseen. Ne on myös otettava projektin seurantaan esimerkiksi vertaamalla valmiskomponentin hintaa ja integroimiseen tarvittavan aikaa ja kustannuksia saman toiminnallisuuden toteuttamiseen arvioituun aikaan ja investointeihin.

Valmiin komponentin toimittaja sitoutuu toimittamaan komponentin dokumentointineen ja mahdollisesti tarvittavine muutoksineen sovittuna ajankohtana.

Prosessien määrittelyt

Jotta valmiskomponenteilla saavutetaan suunniteltu hyöty, on niiden sekä toimittava määritysten mukaisesti että oltava laadultaan riittävän korkeatasoisia. Lisäksi tilaaja saattaa vaatia komponenttiin joitakin muutoksia. Näiden seurantaa ja toimenpiteitä varten on sovittava prosessit, joissa toimittaja on mukana. Toimittajan on sitouduttava osallistumaan tähän prosessiin ja tekemään tarvittavat toimenpiteet poikkeustilanteissa. Nämä prosessit on myös hyvä pitää erillisinä muusta projektin toiminnasta.

Riskien ja ongelmien hallinta

Valmiskomponenteilla ei kuitenkaan aina saavuteta suunniteltuja etuja, joten toimittajan ja asiakkaan kesken on sovittava valmiit menettelyt riskien ja ongelmien hallintaan. Asiakkaalla tulisi olla sekä varasuunnitelmat valmiskomponenttien korvaamiseksi, että toimittajan kanssa sovitut kompensaatiotavat, jos komponentti ei toimi tilatulla tavalla. Toimittajalla tulee puolestaan olla mekanismit reagoida vikailmoituksiin.

Asiakas sitoutuu ilmoittamaan vikatilanteista ajoissa ja toimittaja sitoutuu reagoimaan ilmoituksiin sovitulla tavalla.

5.1.7 Uudelleenkäyttö

Perinteisissä ohjelmistoprojekteissa syntyy usein ohjelmiston osia, joita on mahdollista hyödyntää muissa projekteissa. Kuitenkin varsin harvoin näiden uudelleenkäytettävien komponenttien tuottaminen suunnitellaan ja toteutetaan siten, että niistä olisi mahdollisimman suuri hyöty seuraavissa projekteissa. Usein uudelleenkäytettävyys lepää sen henkilön varassa, joka suunnitteli ja toteutti komponentin. Jos hän ei ole jatkoprojekteissa käytettävissä, ei komponenttejakaan voida käytännössä käyttää.[Fra94]

Olio-ohjelmoinnin yleistyminen on tuonut uudelleenkäytettävien komponenttien tuottamisen yhdeksi tärkeimmistä projektien tavoitteista. Varsinkin toimittaja, mutta myös asiakas suunnittelevat saavansa niistä hyötyä myös jatkoprojekteissa, ja niiden tuottamiselle asetetaan tavoitteita ja vastuuta.[Lim94]

Kun uudelleenkäytettäviä komponentteja tuotetaan projektin sivutuotteena niitä voi syntyä joko valmista ohjelmistoa käyttävälle organisaatiolle tai sille osapuolella, joka ohjelmiston toimitti. Sitoumuksiin vaikuttaa paljon se, mille osapuolelle komponentit jäävät. [Faf94]

Motiivitekijät

Tärkein motiivi uudelleenkäytettävien komponenttien tuottamiselle on luonnollisesti kerran tehdyn investoinnin hyödyntäminen mahdollisimman monessa hankkeessa. Tämä tavoite saavutetaan parhaiten, kun se on kirjattu projektisuunnitelmaan ja -sopimukseen, ja projektista vastaavat henkilöt sitoutuvat siihen.

Vaikka komponenttien tuottaminen ei olisikaan projektin päätavoitteita, se tulisi määritellä jo sopimusta tehtäessä, jotta esim. tekijänoikeudelliset seikat voidaan ottaa huomioon. Varsinkin jos toimittaja aikoo käyttää projektissa syntyneitä komponentteja muille asiakkaille tehtävissä hankkeissa, on toimittajan sitouduttava rajaamaan uudelleenkäyttö sellaisiin komponentteihin ja hankkeisiin, jotka ovat asiakkaalle hyväksyttäviä.

Projektin tavoitteet

Jos projektin päätavoitteena ei ole luoda uudelleenkäytettäviä komponentteja, on niiden luonnin tavoitteet on hyvä asettaa erillään projektin muista tavoitteista. Näin eri tavoitteet voidaan priorisoida, ja projektihenkilöt tai alihankkijat saadaan sitoutettua niihin halutulla tavalla. Usein komponenttien tuottaminen voidaan saattaa loppuun varsinaisen ohjelmistoprojektin päätyttyä, jotta koko projektin päätavoitteita ei vaaranneta.

Komponenttien luonnille asetettavat tavoitteet sisältävät tavallisesti komponentin tyypit, vakiorajapintojen käytön, muodollisen dokumentoinnin, ja aikataulun. Näille kaikille on syytä asettaa omat tavoitteensa.

Prosessien määrittelyt

Jos komponenttien tuottaminen ei ole projektin pääasiallisia tavoitteita, on tuottamisen hallintaan luotava erilliset prosessit, jotta tavoitteiden seuranta

saadaan pidettyä erillisenä muusta projektista. Muussa tapauksessa seuranta on liitettävä osaksi projektin hallintaa.

Riskien ja ongelmien hallinta

Komponenttien tuotanto on mahdollisuuksien mukaan eristettävä varsinaisesta ohjelmistoprojektista, jotta siinä mahdollisesti syntyvät ongelmatilanteet ja niiden käsittely saadaan eristettyä, eivätkä ne vaaranna projektin tavoitteiden täyttymistä.

5.1.8 Teknologioiden tuntemus

Varsinkin tilattaessa ohjelmistoja ohjelmistoprojekteihin erikoistuneelta toimittajalta, ei teknologioiden tuntemusta haluta ottaa huomioon projektista sovittaessa. Teknologioiden tuntemus katsotaan olevan toimittajan asia, jota ei tarvitse tai edes kuulu huomioida. Usein ohjelmistoprojekteissa kuitenkin mennään uusille teknologian alueille, koska uudella ohjelmistolla halutaan saavuttaa merkittävää kilpailuetua. Tämä voi käytännössä tarkoittaa että kaikkia käytettäviin teknologioihin liittyviä riskejä ei voida arvioida tarkkaan, projektin tavoitteita joudutaan tarkistamaan sen edetessä valittujen teknologioiden ominaisuuksien tuntemuksen kasvaessa, ja projektiin osallistuvien on opeteltava uusia asioita samalla kun he työskentelevät projektissa.

Omat merkittävät haasteensa tuovat myös eri osapuolten teknologioiden tuntemuksen erot: jos asiakkaalla ei ole tarpeeksi selkeää kuvaa valittujen teknologioiden mahdollisuuksista, hän voi odottaa lopputuloksesta jotain sellaista, mitä toimittaja ei osannut ennakoida. Tällaisessa tilanteessa ongelmatilanteiden vaara on suuri, koska eri osapuolilla on erilaiset käsitykset projektin tavoitteista.

Motiivitekijät

Valittujen teknologioiden tuntemus vaikuttaa sitoumuksiin erityisesti motiivitekijöiden ja projektin tavoitteiden osalta. On todennäköistä, että projektin perusta ja ohjelmistolla tavoiteltu liiketoiminnan hyöty ovat epärealistisia, jos osapuolet eivät tunne riittävän hyvin teknologian mahdollisuuksia ja riskejä.

Erityisesti jos teknologian tuntemuksessa on suuria eroja asiakkaan ja toimittajan välillä, asiakas saattaa odottaa toimittajan sitoutuvan epärealistisiin tavoitteisiin tai toimittaja antaa lupauksia, joita se ei pysty pitämään. Sekä toimittajan että asiakkaan on tärkeää pyrkiä tunnistamaan epävarmuustekijät teknologian tuntemuksessa, jotta molemmat pystyvät sitoutumaan projektin motiivitekijöihin.

Jos projektissa halutaan soveltaa jotain uutta teknologiaa, on tunnistettava siihen liittyvät riskit ja sovellettava sitoumukset sen mukaan. Uusien teknologioiden tai tuotteiden mukanaan tuomat riskit liittyvät toisaalta niiden mahdollisiin lastentauteihin tai puutteisiin ja toisaalta siihen, että niiden mahdollisuuksia ja rajoituksia ei tunneta hyvin.

Jos projektin tavoitteena on erityisesti kokeilla tai opetella uutta teknologiaa, se on kirjattava nimenomaan projektin motiivitekijäksi. Silloin

projektin vaatimat opettelut ja tavoitteiden epävarmuudet ovat helpommin perusteltavissa ja arvotettavissa.

Projektin tavoitteet

Kun projektissa halutaan soveltaa teknologioita, joita ei uutuutensa takia vielä täysin tunneta, on vaarana että projektilla tavoitellaan asioita, joita ei ole realistisesti mahdollista tehdä. Projektin tavoitteet saattavat olla ristiriidassa teknisten mahdollisuuksien kanssa, ja osapuolet sitoutuvat mahdottomiin asioihin.

Uusien teknologioiden opetteluun tarpeellisuus onkin otettava huomioon asetettaessa projektin tavoitteita. Aikatauluissa on oltava joustovaraa, ja tavoiteltavat ominaisuudet ja toiminnallisuudet ovat määriteltävä joustavasti. Useimmiten paras tapa ottaa nämä joustovarot huomioon on vaiheistaa projekti siten, että kaikkia aikatauluja ja toiminnallisia tavoitteita ei aseteta heti projektin alussa, vaan vasta sen jälkeen kun uusien teknologioiden on ehditty perehtyä.

Jos opettelu on keskeisellä sijalla projektissa, sille voidaan jopa asettaa omat tavoitteensa. Näin sitäkin päästään seuraamaan vastaavasti kuin projektin muita tavoitteita.

Uusien asioiden opettelu vaatii myös aina investointeja. Henkilöiden aikaa kuluu opiskeluun, opiskelumateriaalia on hankittava, kursseille on osallistuttava jne. Kun opettelu on asetettu omaksi tavoitteekseen ja osapuolien siitä saama hyöty on tunnistettu, on näille kuluille myös helpompi löytää maksaja.

Prosessien määrittelyt

Jos projektissa on tarkoitus opetella uutta, on olennaisinta seurata siitä aiheutuvia kustannuksia. Kursseihin ja materiaaleihin liittyvät kulut ovat selkeästi opetteluun kuluja, kun taas opiskeluun käytetyn ajan aiheuttamat kulut eivät ole, joten opetteluun käytetyn ajan ja kulujen seurantaan on luotava oma prosessinsa. Muilta osin uuden opettelulla ei ole suurta vaikutusta projektin prosesseihin.

Riskien ja ongelmien hallinta

Uuden opettelusta aiheutuneet riskit on tunnistettava mahdollisimman varhaisessa vaiheessa, mieluiten jo projektista sovittaessa. Pääasiassa nämä riskit liittyvät projektin aikatauluihin ja toiminnallisuuteen. On mahdollista, että jotain keskeistä toiminnallisuutta ei olekaan mahdollista toteuttaa projektin aikataulun puitteissa, ja tavoitteita tai yhteisiä oletuksia joudutaan muuttamaan. Äärimmäisessä tapauksessa koko projekti joutuu umpikujaan, eikä sitä voida viedä loppuun.

	<i>Motiivitekijät</i>	<i>Projektin tavoitteet</i>	<i>Prosessien määrittelyt</i>	<i>Riskien ja ongelmien hallinta</i>
<i>Projektin koko</i>	- Vaikutukset liiketoiminnalle - Projektin strateginen merkitys?	- Vaiheistus - Jakaminen osaprojekteihin	- Hallinnan työn määrä - Prosessien määrittelyn tärkeys - Varasuunnitelmat	- Riskien ja ongelmien eristäminen - Riskien sietokyky
<i>Osapuolten lukumäärä</i>	- Asiakkaan motiivit - Toimittajien motiivit	- Tavoitteiden jakaminen - Kenellä kokonaistavoite?	- Toimittajien omat prosessit - Yhteiset prosessit	- Riskien jako - Kenellä pääriski? - Ongelmien kommunikointi
<i>Ulkopuolisia toimittajia?</i>	- Asiakkaan ja sisäisen toimittajan yhteiset motiivit - Ulkopuolisen toimittajan motiivit	- Laskutuksen sitominen tavoitteisiin - Entä sisäinen toimittaja?	- Laskutuksen seuranta	- Riskien jako ulkopuolisen kanssa - Entä sisäinen toimittaja?
<i>Aikataulu</i>	- Miksi tietty aikataulu?	- Tuotosten ajoittaminen - Projektin ajankohta	- Aikataulun seuranta - Muutokset aikatauluun	- Kriittisen polun riskit - Ajoitusriskit
<i>Liiketoimintamalli</i>	- T&M, kiinteä hinta, tms - Utility model tai profit sharing	- Laskutuksen sitominen tavoitteisiin - Tavoitteiden ylitys	- Laskutuksen vaiheet - Osalaskutus - Vaikutus kassavirtaan	- Riskien rahallinen arvo - Taloudellisten seurausten jako
<i>Valmis-komponenttien käyttö</i>	- Mitä etua asiakkaalle - Mitä etua toimittajalle	- Kustannusten vertailu - Vaikutus aikatauluun	- Laadun ja sopivuuden varmistus	- Varasuunnitelma - Kompensaatiot vikatilanteissa
<i>Uudelleen-käyttö</i>	- Kenelle jää komponenttien omistus - Mitä etua komponenteilla halutaan saada	- Erilliset tavoitteet - Tavoitteiden priorisointi	- Erillinen prosessi ja seuranta	- Komponenttien tuotannon eristäminen
<i>Teknologian tuntemus</i>	- Onko tavoite opetella uutta? - Teknologian kokeilu?	- Selvitysvaihe - Oma tavoite opettelulle - Investoinnit opetteluun	- Lisäkulujen seuranta	- Valmius suuriin muutoksiin - Projektin peruuttaminen

Taulukko 3 Tekijät sitoumusalueittain

5.2 Tekijöiden vaikutus eri tyyppisissä projekteissa

Projekteissa, joiden tavoite on luoda uusia ohjelmistoja, tehdään laajemmat sitoumukset kuin muun tyyppisissä projekteissa. Suurin syy tähän ovat sekä toimittajan että asiakkaan suuremmat riskit, jotka puolestaan ovat seurausta tämän projektityypin lukuisista epävarmuustekijöistä.

Toisin kuin valmisohjelmistojen projekteissa, tavoitteena olevasta ohjelmistosta ei ole merkittävää osaa valmiina olemassa vaan lähes kaikki joudutaan suunnittelemaan, luomaan ja testaamaan. Lisäksi tavoitteena oleva ohjelmisto on todennäköisemmin ainut laatuaan, toisin kuin valmisohjelmistoja sovellettaessa. Konsultointiprojekteihin verrattuna toimittajan panostus on merkittävästi suurempi, ja useimmiten laskutus on myös sidottu tuotoksiin, joten toimittajalla on samalla myös suurempi taloudellinen riski.

Konsultointiprojekteissa toimittajan on tarkoitus toimia projektissa neuvoa-antavassa ja ohjaavassa roolissa ohjaamassa muiden osapuolten henkilöitä, tuottamatta itse varsinaista ohjelmistoa. Ohjelmistoa tuottavat henkilöt eivät ole muodollisessa vastuussa konsulteille, eikä konsultoivalla osapuolella ole sopimusteknisesti suuria velvollisuuksia tuotettavan ohjelmiston suhteen, mutta todellisuudessa konsulteilla on suuri implisiittinen vastuu.

Tämä vastuun ja toimivallan epäsuhta on usein suurimpia ongelmia projekteissa, joissa toimittaja on konsultoivassa roolissa. Toimittajan sitoumukset projektin tavoitteiden suhteen ovat heikommat kuin muissa projektityypeissä, joten motiivitekijöiden merkitys korostuu.

Valmisohjelmistoja sovellettaessa sitoumukset ovat jossain määrin samantapaiset kuin uusia ohjelmistoja luodessa, mutta vähemmän laajat. Tavallisesti toimittaja on soveltanut samaa valmisohjelmistoa jo muissa hankkeissa, joten projektin riskit tunnetaan suhteellisen hyvin. Lisäksi merkittävä osa ohjelmistoa on jo olemassa useimmiten jopa tuettuna tuotteena, joten projektissa kohdattuihin vaikeuksiin saadaan apua myös projektin ulkopuolisilta tahoilta.

	<i>Uuden ohjelmiston luominen</i>	<i>Konsultointi</i>	<i>Valmisohjelmiston soveltaminen</i>
<i>Projektin koko</i>	<ul style="list-style-type: none"> - Strateginen merkitys - Vaiheistus - Jakaminen osaprojekteihin - Prosessien määrittely - Riskien eristäminen - Riskien sietokyky 	<ul style="list-style-type: none"> - Vaikutukset liike-toiminnalle - Strateginen merkitys - Prosessien määrittely 	<ul style="list-style-type: none"> - Vaikutukset liike-toiminnalle - Strateginen merkitys - Vaiheistus - Jakaminen osaprojekteihin - Riskien eristäminen
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none"> - Asiakkaan motiivit - Tavoitteiden jakaminen - Kenellä kokonaistavoite - Yhteiset prosessit - Riskien jako - Kenellä pääriski? 	<ul style="list-style-type: none"> - Asiakkaan motiivit - Muiden toimittajien motiivit - Kenellä kokonaistavoite - Yhteiset prosessit - Ongelmien kommunikointi 	<ul style="list-style-type: none"> - Tavoitteiden jakaminen - Kenellä kokonaistavoite - Yhteiset prosessit - Riskien jako - Ongelmien kommunikointi
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Laskutuksen seuranta - Riskien jako 	<ul style="list-style-type: none"> - Ulkopuolisen toimittajan motiivit 	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Riskien jako
<i>Aikataulu</i>	<ul style="list-style-type: none"> - Tuotosten ajoittaminen - Projektin ajankohta - Aikataulun muutokset - Kriittisen polun riskit 	<ul style="list-style-type: none"> - Miksi tietty aikataulu? - Aikataulun seuranta ja muutokset - Kriittisen polun riskit 	<ul style="list-style-type: none"> - Tuotosten ajoittaminen - Projektin ajankohta - Kriittisen polun riskit - Ajoitusriskit
<i>Liiketoimintamalli</i>	<ul style="list-style-type: none"> - T&M, kiinteä hinta, profit sharing, tms.? - Laskutuksen sitominen tavoitteisiin - Tavoitteiden ylitys - Laskutuksen vaiheet - Riskien rahallinen arvo - Taloudellisten seurausten jako 	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin 	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Laskutuksen vaiheet - Osalaskutus - Riskien rahallinen arvo
<i>Valmis-komponenttien käyttö</i>	<ul style="list-style-type: none"> - Mitä etua asiakkaalle - Mitä etua toimittajalle - Kustannusten vertailu - Vaikutukset aikatauluun - Laadun ja sopivuuden varmistus - Varasuunnitelma 	<ul style="list-style-type: none"> - Mitä etua asiakkaalle - Laadun ja sopivuuden varmistus - Varasuunnitelma 	<ul style="list-style-type: none"> - Kustannusten vertailu - Vaikutus aikatauluun - Laadun ja sopivuuden varmistus - Varasuunnitelma - Kompensaatiot vikatilanteissa
<i>Uudelleenkäyttö</i>	<ul style="list-style-type: none"> - Kenelle jää komponenttien omistus - Erilliset tavoitteet - Tavoitteiden priorisointi - Erillinen prosessi ja seuranta - Komponenttien tuotannon eristäminen 	<ul style="list-style-type: none"> - Mitä etua komponenteilla haetaan? 	<ul style="list-style-type: none"> - Kenelle jää komponenttien omistus - Erilliset tavoitteet - Tavoitteiden priorisointi - Erillinen prosessi ja seuranta - Komponenttien tuotannon eristäminen

	<i>Uuden ohjelmiston luominen</i>	<i>Konsultointi</i>	<i>Valmisohjelmiston soveltaminen</i>
<i>Teknologian tuntemus</i>	<ul style="list-style-type: none"> - Onko tavoite opetella uutta? - Teknologian kokeilu? - Selvitysvaihe - Oma tavoite opettelulle - Lisäkulujen seuranta - Valmius suuriin muutoksiin - Projektin peruuttaminen 	<ul style="list-style-type: none"> - Onko tavoite opetella uutta? - Teknologian kokeilu? - Valmius suuriin muutoksiin 	<ul style="list-style-type: none"> - Teknologian kokeilu? - Selvitysvaihe - Lisäkulujen seuranta - Valmius suuriin muutoksiin - Projektin peruuttaminen

Taulukko 4 Tekijöiden vaikutus projektityypeittäin

6 Tutkittavien projektien analyysi

Tässä luvussa analysoidaan tutkimuksen kohteena olleet kuusi projektia. Analysointi suoritetaan vertaamalla havainnoituja sitoumuksia edellisessä luvussa esiteltyihin suosituksiin projektityypittäin.

6.1 Projekti A

Tässä projektissa toimitettiin suurelle internet- operaattorille muokattava perusta portaali-järjestelmiä varten. Operaattorin on tarkoitus muokata tämä järjestelmä omille asiakkailleen valmiiksi internet-portaaliksi muokkaamalla sen ulkoasua ja toiminnallisuutta, sekä rakentamalla sen pohjalta uusia sovelluksia, joita käytetään portaalin sisällä.

Projektissa toimittajan vastuulla oli koko portaalin suunnittelu ja toteutus alkaen järjestelmäalustan arkkitehtuurista aina sovelluksien rakentamiseen tarkoitettujen oliokirjastojen luomiseen. Tässä tutkimuksessa keskityttiin projektin ohjelmisto-osuuteen, ja jätettiin järjestelmäalustan arkkitehtuurin suunnittelun ja toteutuksen osuudet käsittelemättä, koska ne eivät sisältäneet ohjelmistojen kehitystä.

Osa hankkeesta oli valmisohjelmistojen soveltamista, mutta suurimman osan projektin sisällöstä muodosti oliokirjastojen suunnittelu ja luominen. Valmisohjelmistojen muokkaaminen tarkoitti käytännössä portaali- ja kalenteripalvelujen muokkaamista asiakkaan määritysten mukaisiksi. Oliokirjastot puolestaan luotiin niitä lisäsovelluksia varten, joita operaattori tekee tai teettää portaali-ohjelmiston lisäksi omia asiakkaitaan varten heidän erityistarpeisiinsa. Tällaisia sovelluksia ovat esimerkiksi projektinhallinta-sovellus PK-sektorin käyttöön, tai sovellus henkilökohtaisen terveystiedon ylläpitoon ja seurantaan.

Tämä projekti käsiteltiin uusien ohjelmistojen luokassa.

Mikä meni hyvin?

Toimittajan kannalta parasta tässä projektissa oli sen aloittaminen lähes kokonaan puhtaalta pöydältä. Suunnittelun lähtökohtana olivat vain asiakkaan tarpeet, eivät mitkään olemassaolevat järjestelmät tai sovellukset. Tämä salli toimittajalle varsin suuren liikkumavaran hankkeen toteuttamiseen.

Projekti oli asiakkaan liiketoiminnalle tärkeä, koska sen avulla asiakas pystyi laajentamaan omaa liiketoimintaansa. Tämän ansiosta asiakas oli valmis panostamaan aikaa ja resursseja, mutta nykyinen liiketoiminta ei kuitenkaan vaarantunut.

Asiakkaan tarpeet ja toimittajan tarjonta sopivat hyvin yhteen. Todennäköisesti juuri tästä syystä projektin lopputulos oli molempia osapuolia tyydyttävä.

Mikä meni huonosti?

Projekti oli ajallisesti pitkä, kestäen noin 2 vuotta ensimmäisistä hahmotteluista asiakkaan kanssa projektin viimeisen vaiheen toteuttamiseen. Tästä syystä asiakkaan organisaatio ja tavoitteet muuttuivat projektin kestäessä useaan otteeseen. Tämä teki projektista osittain liikkuvan kohteen.

Lisäksi asiakkaan organisaation eri osilla oli erilaisia odotuksia projektin suhteen.

Toimittajan organisaatio kehittyi myös projektin aikana voimakkaasti, ja projektipäällikköä jouduttiin vaihtamaan kaksi kertaa. Nämä vaihdot aiheuttivat epäjatkuvuuskohtia projektin kulussa.

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Projektin koko</i>	<ul style="list-style-type: none"> - Strateginen merkitys - Vaiheistus - Jakaminen osaprojekteihin - Prosessien määrittely - Riskien eristäminen - Riskien sietokyky 	<ul style="list-style-type: none"> - Strateginen merkitys ei kirjattu - Vaiheistus määritelty - Prosessit määritelty - Riskianalyysi tehty 	<ul style="list-style-type: none"> - Suuri projekti - Suuri strateginen merkitys asiakkaalle
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none"> - Asiakkaan motiivit - Tavoitteiden jakaminen - Kenellä kokonaistavoite - Yhteiset prosessit - Riskien jako - Kenellä pääriski? 	<ul style="list-style-type: none"> - Tavoitteiden jakaminen kirjattu - Kokonaistavoite kirjattu - Yhteiset prosessit kirjattu - Riskien jako ja pääriskiä ei kirjattu 	<ul style="list-style-type: none"> - Toimittaja päävastuussa asiakkaalle, käyttää alihankkijoita - Merkittävä osa riskeistä toimittajalla
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Laskutuksen seuranta - Riskien jako 	<ul style="list-style-type: none"> - Laskutus sidottu tavoitteisiin - Laskutuksen seuranta määritelty - Riskien jakoa ei määritelty 	<ul style="list-style-type: none"> - Toimittaja sitonut laskutuksen alihankkijoiden tavoitteisiin
<i>Aikataulu</i>	<ul style="list-style-type: none"> - Tuotosten ajoittaminen - Projektin ajankohta - Aikataulun muutokset - Kriittisen polun riskit 	<ul style="list-style-type: none"> - Tuotosten ajoittaminen määritelty - Projektin ajankohtaa ei määritelty - Aikataulun muutosprosesseja ja kriittisen polun riskejä ei määritelty 	<ul style="list-style-type: none"> - Projektin aloitus siirtyi useasti asiakkaan strategiamuutosten takia
<i>Liiketoimintamalli</i>	<ul style="list-style-type: none"> - T&M, kiinteä hinta, profit sharing, tms.? - Laskutuksen sitominen tavoitteisiin - Tavoitteiden ylitys - Laskutuksen vaiheet - Riskien rahallinen arvo - Taloudellisten seurausten jako 	<ul style="list-style-type: none"> - Laskutus sidottu osatavoitteisiin - Tavoitteiden ylitys määritelty - Laskutuksen vaiheet määritelty - Riskianalyysi tehty, ai vaikutusta liiketoiminnan malliin 	<ul style="list-style-type: none"> - Laskutus päivähinnalla vaiheiden valmistuttua - Palkkio arvioiden alituksesta

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Valmiskomponenttien käyttö</i>	<ul style="list-style-type: none"> - Mitä etua asiakkaalle - Mitä etua toimittajalle - Kustannusten vertailu - Vaikutukset aikatauluun - Laadun ja sopivuuden varmistus - Varasuunnitelma 	<ul style="list-style-type: none"> - Valmiskomponenttien käyttö määritelty - Edut ei määritelty - Vaikutukset aikatauluun määritelty - Laadun ja sopivuuden varmistus määritelty - Ei varasuunnitelmaa 	<ul style="list-style-type: none"> - Valmiiden ohjelmistojen ja komponenttien käyttö tärkeä tavoite
<i>Uudelleenkäyttö</i>	<ul style="list-style-type: none"> - Kenelle jää komponenttien omistus - Erilliset tavoitteet - Tavoitteiden priorisointi - Erillinen prosessi ja seuranta - Komponenttien tuotannon eristäminen 	<ul style="list-style-type: none"> - Komponenttien omistus ei määritelty - Ei erillisiä tavoitteita, prosessia tai seurantaa - Komponenttien tuottaminen kirjattu 	<ul style="list-style-type: none"> - Komponenttien tuottaminen kirjattu vain suunnitteluperiaatteena - Komponenttien tuottaminen yksi tärkeistä tavoitteista..
<i>Teknologioiden tuntemus</i>	<ul style="list-style-type: none"> - Onko tavoite opetella uutta? - Teknologian kokeilu? - Selvitysvaihe - Oma tavoite opettelulle - Lisäkulujen seuranta - Valmius suuriin muutoksiin - Projektin peruuttaminen 	<ul style="list-style-type: none"> - Ei uuden teknologian opettelua tai kokeilua - Asiakkaan henkilöstön perehdyttäminen kirjattu 	<ul style="list-style-type: none"> - Uuden teknologian selvittäminen ei varsinainen tavoite

Tämän projektin suurin haaste alussa oli asiakkaan muuttuva organisaatio ja sen mukana muuttuvat tavoitteet. Kun nämä stabiloituivat, projekti saatiin etenemään normaalisti. Nämä muutokset pakottivat kirjaamaan myös asiakkaan korkean tason tavoitteet, koska se oli tärkein väline näiden muutosten hallintaan.

Toinen projektin haasteista oli sen avainhenkilöiden osittainen vaihtuminen, koska projekti oli ajallisesti pitkä. Korkean tason tavoitteiden ja riskien kirjaaminen auttoivat myös näissä muutoksissa.

6.2 Projekti B

Tämän projektin perimmäinen tarkoitus oli selvittää Java-teknologioiden mahdollisuuksia finanssisektorin asiakkaan käyttöön.

Esimerkkisovellukseksi valittiin luotonantosovellus, koska se oli vain asiakkaan sisäisessä, rajallisessa käytössä, eikä pilotin kehitys ollut asiakkaan liiketoiminnan kannalta kriittinen. Sovelluksen oli tarkoitus olla aluksi vain asiakkaan konttorihenkilöstön käytössä, ja jatkossa siitä oli tarkoitus tehdä versio verkkopalvelussa käytettäväksi.

Projekti oli tyypiltään uuden ohjelmiston kehittämistä. Sovelluksen kehityksessä käytettiin geneerisiä valmiskomponentteja lähinnä käyttöliittymän ja tietoliikenneyhteyksien luontiin.

Tämä projekti otettiin tutkimukseen mukaan ennen kaikkea siksi, että sen toteutus ei ollut onnistunut, eikä tavoitteena ollut ohjelmistoa saatu lopullisena versiona toteutettua. Projektin aikana kohdattiin useita erilaisia haasteita, joiden seurauksena se lopulta päätettiin lopettaa sovitussa muodossa.

Mikä meni hyvin?

Ohjelmiston kehityksessä päästiin suhteellisen nopeasti toiminnalliseen prototyyppiin, jolla voitiin todentaa ohjelmiston toimivuus.

Mikä meni huonosti?

Projektin aikana kävi ilmi, että toimittajan ja asiakkaan puolelta projektiin osallistuneilla henkilöillä oli varsin erilaiset käsitykset projektin tavoitteista ja toteutettavien ominaisuuksien painoarvoista. Tämä johti erilaisiin tulkintoihin tavoitteiden täyttymisestä, ja sai projektin tavoitteet muuttumaan tekijöiden kannalta jatkuvasti.

Projektissa törmättiin usein myös haasteisiin ohjelmiston teknisen ympäristön suhteen. Ympäristön luvutut ominaisuudet eivät aina olleet käytettävissä, ja ne saattoivat muuttua projektin aikana.

Projektin sopimus oli toimittajan kannalta epäedullinen varsinkin liiketaloudellisilta ehdoiltaan. Projektille oli esimerkiksi sovittu kiinteä hinta, joka oli selvästi alle tällaisesta hankkeesta normaalisti veloittavan kohtuullisen veloituksen.

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Projektin koko</i>	<ul style="list-style-type: none"> - Strateginen merkitys - Vaiheistus - Jakaminen osaprojekteihin - Prosessien määrittely - Riskien eristäminen - Riskien sietokyky 	<ul style="list-style-type: none"> - Strategista merkitystä ei kirjattu - Vaiheistus määritelty - Prosessit määritelty - Riskejä ei määritelty 	<ul style="list-style-type: none"> - Pieni projekti, mutta suuri strateginen merkitys - Sopimus vaiheesta 1: pilotti - Vaihe 2: tuotanto, väljästi määritelty
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none"> - Asiakkaan motiivit - Tavoitteiden jakaminen - Kenellä kokonaistavoite - Yhteiset prosessit - Riskien jako - Kenellä pääriski? 	<ul style="list-style-type: none"> - Motiiveja ei määritelty - Tavoitteet jaettu - Kokonaistavoitteen omistajaa ei määritelty - Riskien jakoa ei määritelty 	<ul style="list-style-type: none"> - 2 osapuolta
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Laskutuksen seuranta - Riskien jako 	<ul style="list-style-type: none"> - Laskutus sidottu tavoitteisiin - Ei riskien jakoa 	<ul style="list-style-type: none"> - 1 ulkopuolinen toimittaja, asiakkaan henkilöt osallistuivat

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Aikataulu</i>	<ul style="list-style-type: none"> - Tuotosten ajoittaminen - Projektin ajankohta - Aikataulun muutokset - Kriittisen polun riskit 	<ul style="list-style-type: none"> - Ajoittaminen, ajankohta määritetty - Muutoksia ei määritetty - Kriittistä polkua ei määritetty 	<ul style="list-style-type: none"> - Aikataulun seuranta ei onnistunut
<i>Liiketoimintamalli</i>	<ul style="list-style-type: none"> - T&M, kiinteä hinta, profit sharing, tms.? - Laskutuksen sitominen tavoitteisiin - Tavoitteiden ylitys - Laskutuksen vaiheet - Riskien rahallinen arvo - Taloudellisten seurausten jako 	<ul style="list-style-type: none"> - Kiinteä hinta - Laskutus sidottu tavoitteisiin - Tavoitteiden ylityksestä ei palkkiota - Riskien rahallista arvoa ei määritetty 	<ul style="list-style-type: none"> - Kiinteä hinta työarvioon yhdistettynä. - Hinta ei oikeassa suhteessa työmäärään - Rahalliset riskit käytännössä toimittajalla
<i>Valmiskomponenttien käyttö</i>	<ul style="list-style-type: none"> - Mitä etua asiakkaalle - Mitä etua toimittajalle - Kustannusten vertailu - Vaikutukset aikatauluun - Laadun ja sopivuuden varmistus - Varasuunnitelma 	<ul style="list-style-type: none"> - Valmiskomponenttien käyttö määritetty - Edut ei määritetty - Vaikutukset aikatauluun ei määritetty - Ei varasuunnitelmaa 	<ul style="list-style-type: none"> - Käytettiin sekä tuotteistettuja komponentteja että itse tuotettuja
<i>Uudelleenkäyttö</i>	<ul style="list-style-type: none"> - Kenelle jää komponenttien omistus - Erilliset tavoitteet - Tavoitteiden priorisointi - Erillinen prosessi ja seuranta - Komponenttien tuotannon eristäminen 	<ul style="list-style-type: none"> - Uudelleenkäyttö kirjattu - Komponenttien omistus määritetty - Ei erillisiä tavoitteita tai seurantaa 	<ul style="list-style-type: none"> - Tuotettavien komponenttien toteutus ja omistus hyvin määritetty
<i>Teknologioiden tuntemus</i>	<ul style="list-style-type: none"> - Onko tavoite opetella uutta? - Teknologian kokeilu? - Selvitysvaihe - Oma tavoite opettelulle - Lisäkulujen seuranta - Valmius suuriin muutoksiin - Projektin peruuttaminen 	<ul style="list-style-type: none"> - Teknologian kokeilu kirjattu - Selvitysvaihe kirjattu - Ei opettelua - Ei lisäkulujen seurantaa - Valmiutta suuriin muutoksiin ja peruuttamiseen ei kirjattu 	<ul style="list-style-type: none"> - Teknologian kokeilu projekti keskeinen tavoite - Asiakkaan henkilöiden koulutus ei tavoitteena

Tämän projektin suurimmat ongelmat, erilaiset käsitykset tavoitteista ja epäsuhtainen liiketoimintamalli, olisi voitu ainakin osittain välttää panostamalla motiivitekijöiden kunnolliseen kirjaamiseen. Näin tavoitteet olisivat tulleet selkeämmiksi eri osapuolille, ja liiketoiminnan malli olisi rakennettu kestävämmäksi.

6.3 Projekti C

Tämä projekti valittiin esimerkiksi hankkeesta, jossa toimittaja toimi integraattorin alihankkijana. Integraattorin asiakas oli julkishallinnossa toimiva organisaatio, joka toimittaa alueellisia terveydenhoidon palveluja. Tällä asiakkaalla oli tarve ohjelmistolle, joka yhdistää eri terveydenhuollon organisaatioiden tietojärjestelmät toisiinsa. Tavoite oli mm. saada asiakkaan terveydentilasta yhtenäinen kuva vaivattomammin kuin tähän asti puhelimitse ja postin välityksellä.

Tässä projektissa toimittajan asiantuntijat toimivat konsultin roolissa osana integraattorin asiakkaalleen työskentelevää projektiryhmää tietyn alueen teknisinä asiantuntijoina.

Mikä meni hyvin?

Toimittajan konsultit onnistuivat ohjaamaan ohjelmiston toteutuksen sellaiseen tekniseen suuntaan, joka arvioitiin hankkeen onnistumisen kannalta parhaaksi.

Lisäksi toimittajan konsulttien rooli oli varsin selkeä, koska integraattori vastasi suurimmaksi osaksi varsinaisesta suhteesta asiakkaaseen.

Mikä meni huonosti?

Toimittajan konsultit joutuivat useaan kertaan toimimaan välittäjinä integraattorin ja asiakkaan keskinäisissä tai sisäisissä kiistoissa. Tämä rooli oli välttämätön, mutta toimittajan kannalta pulmallinen, koska ko henkilöt ovat enemminkin teknisesti kuin poliittisesti orientoituneita. Lisäksi toimittaja halusi pitää neutraalin asenteen muita osapuolia kohtaan, jotta suhteet säilyisivät hyvinä.

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Projektin koko</i>	<ul style="list-style-type: none">- Vaikutukset liike-toiminnalle- Strateginen merkitys- Prosessien määrittely	<ul style="list-style-type: none">- Vaikutuksia liike-toiminnalle ja strategisista merkitystä ei sopimuksessa- Prosessit määriteltä	<ul style="list-style-type: none">- Keskikokoinen projekti, yhteensä 30-40 henkeä- Merkittävä projekti loppuasiakkaalle
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none">- Asiakkaan motiivit- Muiden toimittajien motiivit- Kenellä kokonais-tavoite- Yhteiset prosessit- Ongelmien kommunikointi	<ul style="list-style-type: none">- Asiakkaan motiivit kirjattu- Integraattorin motiivit kirjattu- Kokonaistavoite kirjattu- Yhteiset prosessit määriteltä	<ul style="list-style-type: none">- Kolme osapuolta: toimittaja, integraattori, asiakas- Motiivit kirjattu osin sopimukseen, osin pöytäkirjoihin
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none">- Ulkopuolisen toimittajan motiivit	<ul style="list-style-type: none">- Ei määriteltä erikseen	<ul style="list-style-type: none">- Asiakkaan tekniset henkilöt eivät osallistuneet projektiin

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Aikataulu</i>	- Miksi tietty aikataulu? - Aikataulun seuranta ja muutokset - Kriittisen polun riskit	- Aikataulu sovittu - Seuranta ja muutokset projektipalaverissa - Kriittistä polkua ei määritelty	- Aikataulu joustava - Käsitelty jokaisessa palaverissa
<i>Liiketoimintamalli</i>	- Laskutuksen sitominen tavoitteisiin	- Laskutusta ei sidottu tavoitteisiin	- Yksinkertainen päivä-hintainen malli
<i>Valmiskomponenttien käyttö</i>	- Mitä etua asiakkaalle - Laadun ja sopivuuden varmistus - Varasuunnitelma	- Valmiskomponenttien käyttö sovittu palaverissa - Ei varasuunnitelmaa	- Geneeriset valmiskomponentit - Ei erityisesti tavoite
<i>Uudelleenkäyttö</i>	- Mitä etua komponenteilla haetaan?	- Komponenttien tuottaminen määritelty - Etuja ei määritelty	- Tavoite kirjattu sopimukseen, seurattu palaverissa
<i>Teknologioiden tuntemus</i>	- Onko tavoite opetella uutta? - Teknologian kokeilu? - Valmius suuriin muutoksiin	- Opettelu ja uuden kokeilu kirjattu - Muutosvalmiudet ei kirjattu	- Tavoite kehittää integraattorin henkilöiden osaamista

Osapuolten motiivit kirjattiin jälkikäteen hälventämään projektissa syntyneitä poliittisia kiistoja. Tämä taktiikka onnistui selkeyttämään eri osapuolten roolit ja siten ehkäisemään kiistat.

Kokonaisvastuu toimituksesta oli integraattorilla. Motiivien kirjaaminen selkeytti myös vastuun jakoa osapuolten välillä etenkin asiakkaan edustajille.

6.4 Projekti D

Tämä julkishallinnon asiakas on tottunut tekemään ennakkoluulottomia teknisiä ratkaisuja omissa tietojärjestelmissään, ja usein poikennut merkittävästikin valtavirrasta. Tässä hankkeessa tavoitteena oli kehittää heidän operatiivinen järjestelmänsä vähitellen kokonaan Java teknologian pohjalle.

Toimittajan asiantuntijat toimivat tässäkin projektissa konsultin roolissa, mutta varsinaisen kehitystyön suorittivat tällä kertaa asiakkaan omat sovelluskehittäjät. Suurin haaste tässä projektissa oli asiakkaan sovelluskehitysorganisaation tuoreus. Asiakas oli perustanut kehitysyksikön varta vasten tälle hankkeelle, ja miehittänyt sen lähes kokonaan uusilla henkilöillä. Lisäksi tämä ryhmä vielä sijaitsi eri kaupungissa kuin asiakkaan muu organisaatio.

Mikä meni hyvin?

Konsultit ovat onnistuneet sekä ohjaamaan kehitystä parhaaksi arvioituun tekniseen suuntaan, että kehittämään asiakkaan sovelluskehittäjien tietotaitoja.

Mikä meni huonosti?

Toimittajan kannalta ongelmallista tässä projektissa on ollut se, että asiakas ei ole sitoutunut tiettyyn työmäärään tai laskutukseen, vaan käytti toimittajan konsultteja hyvin vaihtelevassa määrin. Samoin konsulttien roolia ei ole määritelty riittävän tarkkaan, vaan heiltä on odotettu esimerkiksi asiakkaan kehitysryhmästä vastaavalle henkilölle kuuluvia henkilöstöhallinnollisia tehtäviä.

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Projektin koko</i>	<ul style="list-style-type: none"> - Vaikutukset liike-toiminnalle - Strateginen merkitys - Prosessien määrittely 	<ul style="list-style-type: none"> - Vaikutuksia liiketoiminnalle tai strateginen merkitys ei määritelty - Prosesseista määritelty: laskutus, projektin ohjaus 	<ul style="list-style-type: none"> - Keskikokoinen projekti: 2 toimittajan konsulttia, 12 asiakkaan ohjelmoijaa - Strateginen merkitys neuvotteluissa, ei sopimuksessa
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none"> - Asiakkaan motiivit - Muiden toimittajien motiivit - Kenellä kokonais-tavoite - Yhteiset prosessit - Ongelmien kommunikointi 	<ul style="list-style-type: none"> - Roolit määritelty hyvin - Motiivit ei määritelty - Kokonaistavoitteen omistaja määritelty - Yhteiset prosessit määritelty - Ongelmien kommunikointi ei määritelty 	<ul style="list-style-type: none"> - Vain 2 osapuolta, asiakas ja toimittaja
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none"> - Ulkopuolisen toimittajan motiivit 	<ul style="list-style-type: none"> - Ulkopuolisen toimittajan motiivit ei määritelty 	<ul style="list-style-type: none"> - Yksi ulkopuolinen toimittaja - Päävastuu asiakkaalla
<i>Aikataulu</i>	<ul style="list-style-type: none"> - Miksi tietty aikataulu? - Aikataulun seuranta ja muutokset - Kriittisen polun riskit 	<ul style="list-style-type: none"> - Konsulttien aikataulu määritelty - Muita kohtia ei määritelty - Projektin aikataulu neuvotteluissa 	<ul style="list-style-type: none"> - Keskitytty henkilöiden aikatauluihin - Hankkeen aikataulu asiakkaan vastuulla
<i>Liiketoimintamalli</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin 	<ul style="list-style-type: none"> - Laskutusta ei sidottu tavoitteisiin 	<ul style="list-style-type: none"> - Yksinkertainen päivähintainen malli
<i>Valmiskomponenttien käyttö</i>	<ul style="list-style-type: none"> - Mitä etua asiakkaalle - Laadun ja sopivuuden varmistus - Varasuunnitelma 	<ul style="list-style-type: none"> - Perusteknologia määritelty - Valmiskomponenttien käytöstä päätetty projektipalaverissa - Laadun varmistusta, varasuunnitelmaa ei määritelty 	<ul style="list-style-type: none"> - Valmiskomponenttien käyttö ei varsinaisesti tavoitteena - Projektin aikana otettu käyttöön joitakin komponentteja
<i>Uudelleenkäyttö</i>	<ul style="list-style-type: none"> - Mitä etua komponenteilla haetaan? 	<ul style="list-style-type: none"> - Edut ei määritelty - Komponenttien tuottaminen kirjattu 	<ul style="list-style-type: none"> - Komponenttien tuottaminen asiakkaan tavoite!

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Teknologioiden tuntemus</i>	<ul style="list-style-type: none"> - Onko tavoite opetella uutta? - Teknologian kokeilu? - Valmius suuriin muutoksiin 	<ul style="list-style-type: none"> - Tavoite on opetella ja kokeilla uutta, ei kuitenkaan määritelty sopimuksessa - Konsulttien rooli perehtymisessä kuvattu - Käytäntöjen kehittäminen kirjattu 	<ul style="list-style-type: none"> - Yksi asiakkaan tärkeimmistä tavoitteista!

Projektin sopimuksessa ja projektipalavereissa käsiteltiin hyvin vähän uudelleenkäytettävien komponenttien tuottamista ja asiakkaan henkilökunnan kehittämistä, jotka olivat kuitenkin korkealla tasolla asiakkaan tärkeimpiä tavoitteita. Uudelleen käytettävien komponenttien tuottamista käsiteltiin ainoastaan projektin palavereissa. Oletettavasti näin tapahtui, koska projektin johto ja päävastuu olivat asiakkaalla, ja toimittajan edustajat katsoivat tällaisten tavoitteiden olevan suurimmaksi osaksi toimittajan vastuiden ulkopuolella.

Projektin suurimpia haasteita oli kuitenkin juuri asiakkaan henkilöstön kehittäminen, varsinkin kun lähes koko kehitysryhmä oli vasta äskettäin palkattu. Jos henkilöstön kehittämisen tavoitteet ja sitoumukset olisi kirjattu sopimukseen tai vähintään käsitelty palavereissa, olisivat nämä ongelmat saattaneet olla lievempiä.

6.5 ***Projekti E***

Tämä hanke sai alkunsa asiakkaan verkkopalvelun suorituskykyongelmista. Projektissa konsulttoitiin asiakasta ja asiakkaan ohjelmistotoimittajia suorituskykyongelmien selvittämiseksi.

Tämä projekti sijoitettiin tutkimuksessa konsultointi- luokkaan. Siinä myös muokattiin sekä aikaisemmin toisen toimittajan tuottamia ohjelmistoja että asiakkaan käytössä olleita valmisohjelmistoja, mutta tämä työ ei ollut projektin pääasiallinen sisältö.

Mikä meni hyvin?

Toimittajan konsultit onnistuivat löytämään sekä asiakkaan sovelluksessa että järjestelmässä useita kehityskohteita, joihin keskittymällä kokonaisjärjestelmän suorituskykyä voitiin kohentaa merkittävästi. Lisäksi suhde asiakkaaseen toimi hyvin, koska konsultit olivat toimineet tämän asiakkaan kanssa jo pitkään.

Mikä meni huonosti?

Selvitysten edetessä kävi ilmi, että järjestelmän suorituskykyongelmat johtuivat pääosin muista seikoista kuin ohjelmistosta. Koska taas monet näistä seikoista olivat toimittajan konsulttien toimialueen ulkopuolella, ei niihin voitu vaikuttaa ratkaisevasti. Edelleen tästä johtuen saavutetut suorituskyvyn ja varsinkin skaalautuvuuden kohennukset jäivät tavoiteltua pienemmiksi.

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Projektin koko</i>	<ul style="list-style-type: none"> - Vaikutukset liike-toiminnalle - Strateginen merkitys - Vaiheistus - Jakaminen osaprojekteihin - Riskien eristäminen 	<ul style="list-style-type: none"> - Ei määritelty sopimuksessa 	<ul style="list-style-type: none"> - Pieni projekti, mutta suuri strateginen merkitys..
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none"> - Tavoitteiden jakaminen - Kenellä kokonais-tavoite - Yhteiset prosessit - Riskien jako - Ongelmien kommunikointi 	<ul style="list-style-type: none"> - Kokonaistavoitteen omistaja määritelty - Yhteiset prosessit määritelty palaverissa 	<ul style="list-style-type: none"> - 2 osapuolta, asiakas ja toimittaja
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Riskien jako 	<ul style="list-style-type: none"> - Laskutusta ei sidottu tavoitteisiin 	<ul style="list-style-type: none"> - 1 ulkopuolinen toimittaja
<i>Aikataulu</i>	<ul style="list-style-type: none"> - Tuotosten ajoittaminen - Projektin ajankohta - Kriittisen polun riskit - Ajoitusriskit 	<ul style="list-style-type: none"> - Aikataulu ja ajankohta määritelty - Maksimi ajankäyttö määritelty - Riskejä ei määritelty 	<ul style="list-style-type: none"> - Toimittajalla riski maksimajan takia
<i>Liiketoimintamalli</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Laskutuksen vaiheet - Osalaskutus - Riskien rahallinen arvo 	<ul style="list-style-type: none"> - Laskutusta ei sidottu tavoitteisiin - Laskutus ei vaiheistettu 	<ul style="list-style-type: none"> - Yksinkertainen päivähintainen malli
<i>Valmiskomponenttien käyttö</i>	<ul style="list-style-type: none"> - Kustannusten vertailu - Vaikutus aikatauluun - Laadun ja sopivuuden varmistus - Varasuunnitelma - Kompensaatiot vikatilanteissa 	<ul style="list-style-type: none"> - Ei määritelty sopimuksessa 	<ul style="list-style-type: none"> - Valmiskomponenttien käyttö ei tavoitteena
<i>Uudelleenkäyttö</i>	<ul style="list-style-type: none"> - Kenelle jää komponenttien omistus - Erilliset tavoitteet - Tavoitteiden priorisointi - Erillinen prosessi ja seuranta - Komponenttien tuotannon eristäminen 	<ul style="list-style-type: none"> - Komponentoinnin edellytysten tarkastus määritelty - Muuten ei kirjattu 	<ul style="list-style-type: none"> - Ei luotu uusia komponentteja - Tarkastettiin olemassa-olevat komponentit ja menetelmät

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Teknologioiden tuntemus</i>	<ul style="list-style-type: none"> - Teknologian kokeilu? - Selvitysvaihe - Lisäkulujen seuranta - Valmius suuriin muutoksiin - Projektin peruuttaminen 	- Ei määritelty	- Opettelu tai uuden teknologian kokeilu ei tavoitteena

Koska tämä projekti oli varsin pieni ja kestoaltaan lyhyt, sen sopimus oli myös varsin lyhyt, eikä varsinaisia projektipalavereita ollut kuin kolme. Pienuudestaan huolimatta projekti oli strategiselta merkitykseltään suuri sekä asiakkaalle että toimittajalle, joten varsinkin korkean tason sitoumukset olisi todennäköisesti kannattanut kirjata paremmin. Projektin suorittamiseen osallistuvat henkilöt kuitenkin ymmärsivät sen merkityksen, vaikka sitä ei oltu kirjattu mihinkään.

6.6 Projekti F

Tämä projekti oli itse asiassa jatkohanke Projekti A:lle, ja se aloitettiin noin 2 vuotta Projekti A:n jälkeen. Se otettiin mukaan tähän tutkimukseen ennen kaikkea siksi, että havaittaisiin miten välissä kulunut aika, osapuolten suhteiden kehittyminen tai muu kehitys vaikuttanut sitoumusten käsittelyyn. Toisaalta vaikka kyseessä oli itsenäinen projekti, monet osapuolten keskinäisistä asioista oli jo sovittu kertaalleen Projekti A:ssa.

Projekti F:n sisältö oli operaattorin omalle loppuasiakkaalle suunnatun sovelluksen kehittäminen Projekti A:ssa kehitetyn teknologian pohjalta. Asiakkaalle toimitettava sisältö oli siis valmis sovellus, joten tämä projekti katsotaan kuuluvan kokonaan uuden ohjelmiston tuottamiseen.

Mikä meni hyvin?

Koska tämä projekti oli toisen hankkeen jatkoprojekti, sopimusneuvottelut sujuivat varsin hyvin. Osapuolet tunsivat toisensa jo varsin hyvin, joten monet sitoumuksista olivat implisiittisiä. Vaikka projektin käynnistäminen venyi asiakkaan aikatauluista johtuen, se käynnistyttyään eteni suunnitelman mukaan.

Mikä meni huonosti?

Juuri koska projektissa ei katsottu tarpeelliseksi määritellä kaikki asioita uudestaan, se aloitettiin varsin kevyeltä pohjalta. Tämä keveys olisi ollut jossain toisessa hankkeessa liian suuri riski, mutta tässä sen vaikutus katsottiin riittävän pieneksi.

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Projektin koko</i>	<ul style="list-style-type: none"> - Strateginen merkitys - Vaiheistus - Jakaminen osaprojekteihin - Prosessien määrittely - Riskien eristäminen - Riskien sietokyky 	<ul style="list-style-type: none"> - Ei määritelty - Prosessit määritelty 	<ul style="list-style-type: none"> - Pieni projekti - Projektissa viitattiin edellisen vaiheen perusmäärittelyyn
<i>Osapuolten lukumäärä</i>	<ul style="list-style-type: none"> - Asiakkaan motiivit - Tavoitteiden jakaminen - Kenellä kokonaistavoite - Yhteiset prosessit - Riskien jako - Kenellä pääriski? 	<ul style="list-style-type: none"> - Ei määritelty sopimuksessa, kirjattiin palaverissa - Kokonaistavoite määritelty 	<ul style="list-style-type: none"> - Vain 2 osapuolta - Prosessit ja riskien jako edellisestä sopimuksesta
<i>Ulkopuolisia toimittajia?</i>	<ul style="list-style-type: none"> - Laskutuksen sitominen tavoitteisiin - Laskutuksen seuranta - Riskien jako 	<ul style="list-style-type: none"> - Laskutus ei sidottu tavoitteisiin 	<ul style="list-style-type: none"> - Päivähintainen malli
<i>Aikataulu</i>	<ul style="list-style-type: none"> - Tuotosten ajoittaminen - Projektin ajankohta - Aikataulun muutokset - Kriittisen polun riskit 	<ul style="list-style-type: none"> - Aikataulu sovittu - Muutoshallinta ei sovittu - Kriittistä polkua ei määritelty 	<ul style="list-style-type: none"> - Aikataulu kriittinen, mutta projekti suora-viivainen
<i>Liiketoimintamalli</i>	<ul style="list-style-type: none"> - T&M, kiinteä hinta, profit sharing, tms.? - Laskutuksen sitominen tavoitteisiin - Tavoitteiden ylitys - Laskutuksen vaiheet - Riskien rahallinen arvo - Taloudellisten seurausten jako 	<ul style="list-style-type: none"> - Laskutus ei sidottu tavoitteisiin - Ylitystä, vaiheita tai riskejä ei määritelty 	<ul style="list-style-type: none"> - Yksinkertainen malli päivähinnittelulla
<i>Valmiskomponenttien käyttö</i>	<ul style="list-style-type: none"> - Mitä etua asiakkaalle - Mitä etua toimittajalle - Kustannusten vertailu - Vaikutukset aikatauluun - Laadun ja sopivuuden varmistus - Varasuunnitelma 	<ul style="list-style-type: none"> - Valmiskomponenttien käyttö kirjattu - Etuja ei kirjattu - Vaikutukset aikatauluun kirjattu palaverissa - Laadun varmistus sovittu palaverissa 	<ul style="list-style-type: none"> - Käytettiin edellisessä projektissa tuotettuja komponentteja
<i>Uudelleenkäyttö</i>	<ul style="list-style-type: none"> - Kenelle jää komponenttien omistus - Erilliset tavoitteet - Tavoitteiden priorisointi - Erillinen prosessi ja seuranta - Komponenttien tuotannon eristäminen 	<ul style="list-style-type: none"> - Omistus sovittu - Ei erillistä prosessia ja seurantaa 	<ul style="list-style-type: none"> - Uusien komponenttien tuottaminen projektin päätavoite

<i>Vaikuttava tekijä</i>	<i>Suositus</i>	<i>Havainnot</i>	<i>Kommentteja</i>
<i>Teknologioiden tuntemus</i>	<ul style="list-style-type: none"> - Onko tavoite opetella uutta? - Teknologian kokeilu? - Selvitysvaihe - Oma tavoite opettelulle - Lisäkulujen seuranta - Valmius suuriin muutoksiin - Projektin peruuttaminen 	<ul style="list-style-type: none"> - Ei opetella eikä kokeilla uutta - Valmiutta muutoksiin tai peruttamiseen ei määritelty 	<ul style="list-style-type: none"> - Käytetään edellisen projektin teknologioita ja osaamista

Tämä projekti toimi hyvänä esimerkkinä hankkeesta, jonka toisensa hyvin tuntevat osapuolet sopivat. Lisäksi koska tämä projekti oli jatkoa toiselle projektille, monet edellisessä projektissa sovitut asiat soveltuivat sellaisenaan tähänkin. Tällaisessa tapauksessa projektilla on hyvät mahdollisuudet onnistua, vaikka siitä muodollisesti sovitaan erittäin kevyesti.

6.7 Yhteenveto

Tutkittavat kuusi projektia luokiteltiin ennalta johonkin kolmesta luokasta sen perusteella, millainen toimittajan siihen tuoma panos oli. Tulosten perusteella näyttää siltä, että tällä luokittelulla saadaan riittävä jako myös erilaisten sitoumusten tekemiseen. Tämä luokittelun lähtökohta on myös käytännössä paras, koska projekti voidaan tunnistaa kuuluvaksi johonkin näistä luokista hyvin alkuvaiheessaan, ennen kuin useimpia muita tietoja siitä on vielä saatavilla. Kun eri projektiluokkien suositukset sitoumuksista ovat tiedossa heti kun sopimusneuvottelut käynnistetään, sitoumusten teossa osataan keskittyä olennaisimpiin alusta asti.

Tätä työtä tehdessä oletettiin ennakolta, että ohjelmiston kehityskaaren mallit vaikuttaisivat merkittävästi sitoumusten tekoon. Näyttää siltä, että vaikka ne vaikuttavat teoriassa, niitä ei käytetä erityisesti välineenä käytännössä sitoumusten hallintaan. Tämä johtuu todennäköisesti siitä, että nykyisessä oliosuuntautuneessa ohjelmoinnissa käytetään RUP tai vastaavia inkrementaalisia malleja niin laajasti, että niistä on tullut kiinteä osa ohjelmistokehitystä.

7 Johtopäätökset ja suositukset

Tämän tutkimuksen perusteella on mahdollista tunnistaa ne tekijät, jotka eniten vaikuttavat erilaisissa projekteissa tehtäviin sitoumuksiin. Useimmat projekteissa hyvin tai vähemmän hyvin menneistä asioista olivat nähtävissä siitä, miten näitä sitoumuksia oli otettu huomioon ja kirjattu projektien sopimukseen ja palaverin pöytäkirjoihin.

Projektin tyyppi vaikuttaa tämän tutkimuksen perusteella tarvittaviin sitoumuksiin. Tässä työssä esitettiin projektien luokittelu, joka soveltuu erityisesti sellaisen organisaation käyttöön, jossa kirjoittaja työskentelee, koska se toimii nimenomaan esitetyn tyyppisissä projekteissa. Toisena laiselle organisaatiolle saattaisi jokin muu jako olla sopivampi. Esimerkiksi jos yrityksellä ei ole tarjota muokattavaa valmisohjelmistoa, vaan se keskittyy tietyn toimialueen uusien ohjelmistojen toimittamiseen, projektien luokittelun perusta voisi olla tuotettavan ohjelmiston tyyppi tai asiakkaan koko.

Toimittajan vastuun laajuudella on sitoumuksiin suuri merkitys. Vastuu on laajimmillaan uusien ohjelmistojen tuottamisessa, ja silloin tehtävät sitoumukset ovat sekä moninaisemmat että voimakkaammat kuin projekteissa, joissa toimitetaan valmisohjelmistoja. Tutkimuksen perusteella näyttää myös siltä, että vaikka puhtaassa konsultointiprojektissa toimittajan virallinen vastuu on varsin kapea, se on todellisuudessa kuitenkin suurempi kuin varsinkin valmisohjelmistoja sovellettaessa.

7.1 Sitoumukset osana myyntiprosessia

Voidaanko sitoumuksilla myös ohjata myyntiprosessia, ja tehdä tarjous houkuttelevammaksi asiakkaalle? Voidaanko tarjousten ja sopimusten laatua toimittajan kannalta nostaa panostamalla sitoumuksiin?

Ottaessaan laajempaa vastuuta toimittaja siirtää riskiä asiakkaalta itselleen. Samalla toimittaja tekee laajempia sitoumuksia, jotka kattavat projektin lisäksi myös asiakkaan liiketoiminnan tavoitteita ja motivaatioita. Tällä tavalla toimittaja voi kehittää suhdettaan asiakkaaseen siten, että hän saavuttaa asiakkaan kumppanin aseman. Samalla asiakaskin on sitoutettava laajemmin, jotta suhteesta saadaan kehittymään tasapuolisen kumppanuus.

Tarjouksen ja sopimuksen laatu sekä toimittajan ja asiakkaan kannalta kohenee, kun molempien osapuolten tekemät sitoumukset saadaan tuotua esiin. Molemmat pystyvät näin sekä arvostamaan että arvottamaan taloudellisesti toisen osapuolen sitoumukset.

7.2 Sitoumukset osana projektinhallintaa

Sitoumusten hallinta alkaa ohjelmistoprojektin myynnistä ja jatkuu projektin hallintaan. Yksi projektin hallinnan suurimmista haasteista on niihin liittyvän suuren tietomäärän hallitseminen, ja sitoumusten hallinta antaa yhdenlaisen näkymän siihen, mitkä seikat ovat olennaisimpia. Sitoumusten hallinta on joka tapauksessa aloitettava jo projektin myyntivaiheessa, koska projektin ollessa käynnissä niistä on huomattavasti vaikeampi sopia.

7.3 Jatkotutkimuksen aiheita

Varsinkin koska sitoumuksia on käsitelty kirjallisuudessa hyvin vähän omana alueenaan, olisi nyt tehdyn tutkimuksen tueksi saatava lisää empiiristä aineistoa. Tätä aineistoa voidaan kerätä samalla menetelmällä kuin tässä työssä tehtiin, mutta käyttäen aineistona suurempaa määrää projekteja. Mielenkiintoinen lisäys empiiriseen materiaaliin olisi ollut projektiin osallistuneiden henkilöiden haastattelut, koska tällä tavalla olisi saatu koottua eri osapuolten näkemys projektista. Tässä työssä esitetty materiaali on kuitenkin yksi näkymä projektin sitoumuksiin.

Koherentin empiirisen materiaalin kokoaminen post-mortem on kuitenkin vaikeaa. Paras materiaali saataisiin jos ohjeistoa kokeiltaisiin käytännössä. Nyt esitetyjä suosituksia tullaan käyttämään kirjoittajan yrityksen jatkoprojekteissa, joten näistä projekteista olisi mahdollista saada tutkimusmateriaalia myös projektin myyntivaiheessa ja sen aikana. Tämän materiaalin kokoaminen ja analysointi joudutaan jättämään tulevaan tutkimukseen.

Eräs kiinnostava näkökulma malliin olisivat eri tekijöiden painotukset. Tässä esitetyssä tutkimuksessa tekijöiden painotuksia ei ollut mahdollista selvittää, osittain siksi että asteikon luominen tekijöiden painoille ei ole olemassa, ja osittain siksi että tällaista materiaalia on vaikea saada koottua tutkittaessa projekteja niiden päättymisen jälkeen.

Liiketaloudelliselta kannalta mielenkiintoinen tutkimusalue olisi myös sitoumusten vaikutus projektin hintaan. Millaisissa tapauksissa asiakkaalle on edullista, että toimittaja ottaa laajan vastuun ja sitoutuu asiakkaan omiin tavoitteisiin ja motivaatioihin? Tämä tutkimus on tässä esitetyn työn ulkopuolella, mutta tarjoaisi joka tapauksessa mielenkiintoisen ja käytännöllisen näkökulman ohjelmistoprojekteihin.

8 Yhteenveto

Projektista sopiminen ja sen hallinta ovat monimutkaisia tapahtumia, joiden onnistunut hallinta vaatii projektihenkilöiltä pitkää kokemusta.

Kokeneimmat henkilöt ovat kuitenkin harvoin käytettävissä. He keskittyvät suurimpien ja haastavimpien hankkeiden hallintaan, ja muut jäävät kokemattomampien henkilöiden hoidettavaksi. Kuitenkin näissä pienemmissäkin hankkeissa voi piillä yhtä suuria riskejä kuin suuremmissa, joten niidenkin onnistuminen on jotenkin varmistettava.

Sitoumusten ohjeistolla on mahdollista helpottaa sekä projektin sopimusten tekemistä että projektin ohjausta. Ne eivät korvaa muita työkaluja kuten vakiosopimuksia ja projektinhallinnan menetelmiä, vaan täydentävät niitä tuomalla projektiin erilaisen näkökulman. Ne auttavat myös hahmottamaan projektin osapuolten väliset suhteet ja vastuut huomattavasti paremmin

Ohjelmistoprojektin menestyksen salaisuuksista keskeisimpiä ovat osapuolten ymmärrys toistensa tavoitteista ja motiiveista, ja projektin monimutkaisuuden riittävä yksinkertaistaminen. Sitoumusten hallinta auttaa näissä molemmissa tavoitteissa ja siten omalta osaltaan nostaa projektin onnistumisen todennäköisyyttä.

Lähdeluettelo

- Abe79* Joichi Abe, Ken Sakamura, Hideo Aiso, An analysis of software project failure, Proceedings of 4th Software Engineering, 1979
- Bec00* Kent Beck, eXtreme Programming explained, Addison-Wesley, 2000
- Boe88* Barry W. Boehm, A Spiral Model of Software Development and Enhancement, IEEE Computer, 1988
- Boe89* Barry W. Boehm, Rony Ross, Theory W Software Project Management: Principles and Examples, IEEE Transactions on Software Engineering, 1989
- Boe98* Barry Boehm, Alexander Egyed, Software requirements negotiation: some lessons learned, IEEE, 1998
- Bra97* Ken Bradley, Understanding Prince2, SPOCE Project Management Ltd, 1997
- Bro95* Frederick P. Brooks, The Mythical Man-Month. 2nd ed., Addison-Wesley, 1995
- Faf94* Danielle Fafchamps, Organisational Factors and Reuse, IEEE Software, 1994
- Fra94* William B. Frakes, Sadahiro Isoda, Success Factors of systematic reuse, IEEE Software, 1994
- Grü01* Paul Grünbacher, Barry Boehm, EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation, Communication of the ACM, 1998
- IEE87* IEEE, IEEE Standard for Software Project Management Plans, std 1058.1-1987, IEEE, 1987
- Jac99* Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Process, IEEE Software, 1999
- Kes01* Keskuskauppakamari, IT2000 Tietotekniikka-alan sopimusehdot, 2001
- Kon97* Jyrki Kontio, The Riskit Method for Software Risk Management, Computer Science Technical Reports, 1997
- Kon98* Jyrki Kontio, Olli Pitkänen and Reijo Sulonen, Towards Better Software Projects and Contracts: Commitment Specifications, 1998
- Kru00* Philippe Kruchten, From Waterfall to Iterative Lifecycle, Rational Software, 2000
- Lap01* M Lappi, Formation Process of Dyadic relationships: Finnish Software Companies Established in USA, Oulu University Press, 2001
- Lim94* Wayne C. Lim, Effects of Reuse on Quality, Productivity and Economics, IEEE Software, 1994

- Man99* Dino Mandrioli, Paul Zeiger, Software Project Failure Lessons Learned, Communications of the ACM, 1999
- Mar00* Robert G. Martin, eXtreme Programming Development through Dialog, IEEE Software, 2000
- McC98* Steve McConnell, Ohjelmistoprojektit - Selviytymisopas, Microsoft Press/ Satku, 1998
- Smi00* John Smith, A Comparison of RUP and XP, Rational Software, 2000
- Voa98a* Jeffrey M. Voas, The Challenges of Using COTS Software in Component-based Development, IEEE Software, 1998
- Voa98b* Jeffrey M. Voas, Certifying Off-the-Shelf Software Components, IEEE Software, 1998
- War01* Juhani Warsta, Contracting in Software Business, Doctoral dissertation, Oulu University Press, 2001